12-8-08 AF/IFW

Attorney Docket No. 15569-0007

ON APPEAL TO THE U.S. PATENT AND TRADEMARK OFFICE

BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Parsons, et al.)		
Appl. No.:	10/526,107)	Examiner:	Jean, Frantz B.
Filing Date:	February 28, 2005)	Art Unit:	2151
Int'l Appl. No.:	PCT/US02/27956)		
Int'l Filing Date:	August 30, 2002)		

Title: **PROXY EMAIL METHOD AND SYSTEM**

CERTIFICATE OF MAILING BY EXPRESS MAIL
"Express Mail" Mail Label Number EM304436050US

Sir:

I hereby certify that the following correspondence is being deposited in the United States Postal Service as Express Mail on the date shown below in an envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450:

1. Letter Accompanying Appellants' Appeal Brief (2 pages);
2. Appellants Brief (19 pages in triplicate);
3. Appendix A (5 pages in triplicate);
4. Appendix B (1 page in triplicate);
5. Appendix C (1 page in triplicate);
6. Appendix D (32 pages in triplicate);
7. Appendix E (17 pages in triplicate);
8. Appendix F (31 pages in triplicate);
9. Check for $2,270.00; and
10. A return receipt postcard.

Dated: Dec. 5, 2008 Suzanne Shields

Suzanne Shields

GALLAGHER & KENNEDY, P.A.
Attorneys at Law
2575 East Camelback Road
Phoenix, AZ 80516-9225
Tel. No. (602) 530-8000

1084868v3

ON APPEAL TO THE U.S. PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Parsons, et al.	)	Art Unit:	2151
Appl. No.:	10/526,107	)	Examiner:	Jean, Frantz B.
Filing Date:	February 28, 2005	)		
Int'l Appl. No.:	PCT/US02/27956	)		
Int'l Filing Date:	August 30, 2002	)		

Title: **PROXY EMAIL METHOD AND SYSTEM**

LETTER ACCOMPANYING APPELLANTS' APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Enclosed are three copies of Appellants' Appeal Brief under 35 U.S.C. § 134(a) in the application identified above.

A four month extension of time fee is requested for the filing of the enclosed and a check for $2,270 is enclosed to cover the extension fee and the Appeal Brief filing fee. In the event of any deficiency, authorization is given to charge the deposit account 070135 of the undersigned for any deficiency or other required fee.

Respectfully submitted,

12/09/2008 HDESTA1 00000011 10526107
02 FC:1254 1730.00 OP

GALLAGHER & KENNEDY

Date: December 5, 2008

By: Thomas D. MacBlain
Reg. No. 24,583
Attorney for Appellant

1963114

Gallagher & Kennedy, P.A.
2575 East Camelback Road
Phoenix, AZ 85016-9225
602-530-8088
tdm@gknet.com

ON APPEAL TO THE U.S. PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Parsons, et al.	)	Art Unit:	2151
Appl. No.:	10/526,107	)	Examiner:	Jean, Frantz B.
Filing Date:	February 28, 2005	)		
Int'l Appl. No.:	PCT/US02/27956	)		
Int'l Filing Date:	August 30, 2002	)		

Title: **PROXY EMAIL METHOD AND SYSTEM**

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' APPEAL BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

STATUTES

OTHER AUTHORITIES

CASES

ON APPEAL TO THE U.S. PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Parsons, et al.	)	Art Unit:	2151
Appl. No.:	10/526,107	)	Examiner:	Jean, Frantz B.
Filing Date:	February 28, 2005	)		
Int'l Appl. No.:	PCT/US02/27956	)		
Int'l Filing Date:	August 30, 2002	)		

Title: **PROXY EMAIL METHOD AND SYSTEM**

APPELLANTS' APPEAL BRIEF

This is Appellant's Brief on Appeal pursuant to 35 U.S.C. § 134(a). The following sections of this Brief are the items set forth in 37 C.F.R. § 41.37(c).

Real Party in Interest (37 C.F.R. § 41.37(c) (1) (i))

The Go Daddy Group, Inc., a corporation under the laws of Arizona and assignee of the inventors Robert R. Parsons, Joshua T. Coffman and Barbara J. Rechterman, is the real party in interest.

Related Appeals and Interferences (37 C.F.R. 41.37(c) (1) (ii))

There are no related appeals or interferences.

Status of Claims (37 C.F.R. § 41.37(c) (1) (iii))

Claims 1 - 26 - rejected.

Claims 1 - 26 are all of the claims present in this application. Each of claims 1 - 26 is under final rejection. The claims on appeal are appended at Appendix A.

Status of Amendments (37 C.F.R. § 41.37(c) (1) (iv))

There has been no amendment of the application subsequent to the final rejection dated March 20, 2008.

Summary of Claimed Subject Matter (37 C.F.R. § 41.37(c) (1) (v))

Without limiting the interpretation of claims in appellants' application, which claims stand on their own, all claims relate to "proxy email" installations, methods or programming where a proxy entity receives email for customers, at customer proxy email addresses, then decides which, if any, email message to send to a customer in an email to the customer's actual email address based on the customer's previously expressed preferences.

The independent claims involved in this appeal are 1, 2, 6 and 11. Again, without limiting the interpretation of the claims to the specific elements of any exemplary embodiment described in the specification, the following explains the claims' subject matter with reference to the specification and drawing figures. References in parentheses are to the drawing figure, page and line number of the appellants' application as filed on February 28, 2005.

Independent Claim 1

The independent claim 1 is directed to a "proxy email computer installation" (block 60 in Fig. 3 or Fig. 7). The installation includes a database in computer memory (as indicated at 194 in Fig. 8b) that associates a customer's identification (which can include the customer's personal contact information shown at 194 in Fig. 8b and as described, for example, at page 10, lines 4 - 20). Also included in the proxy email computer installation is "an email server having an ability to receive a first email message at the customer's email address" (Fig. 9, block 136, page 10, lines 9 - 12).

The proxy email computer installation of claim 1 has "computer executable code on a computer usable medium or media" (represented by the flow chart of Fig. 9, page 10, line 10). This code includes "(i) first programming to retrieve a customer's actual email address from the database upon receipt of the first email message to the customer's proxy email address" (block 138 of Fig. 9, page 10, lines 10 - 13). The code also includes "(ii) second programming to

forward a second email message to the customer's actual email address, wherein the second email message is formed from the first email message" (Fig. 9, block 148 and page 10, lines 32 - 34).

The proxy email computer installation of claim 1 further includes "a connection to a communication link forwarding the second email message to the customer's actual email addresses" (the communication line 150 of Fig. 9, page 11, lines 1 and 2).

Claims dependent from Claim 1

Claims 15 - 18 are dependent claims depending from claim 1 directly or through mesne dependencies.

Claim 15 provides that the email server of claim 1 "is set up to receive all incoming proxy emails to a common catch-all account" (page 10, lines 13 and 14). Claim 15 also provides that "the email computer installation further comprises an email address management system" (Fig. 7, block 60) "that periodically checks for new emails for any customer" (page 10, lines 15 - 17).

Claim 16, dependent from claim 15, provides that "the first programming to retrieve a customer's actual email address includes programming to check email addresses of all addressed and copied parties for a proxy email address being served by the proxy email server whenever a new email is found by the email address management system" (page 10, lines 21 - 23).

Claim 17 is dependent from claim 16 and provides "the second programming comprises programming to copy the content of the first email message into the second email message, the second email message comprising an email message from the proxy computer installation to the customer at the customer's actual email address" (page 10, lines 32 - 34).

Claim 18 depends from claim 17 and further provides "programming in computer executable code on a computer usable medium or media to effect deletion of an email once addresses of all addressed and copied parties have been checked for proxy email addresses of customers and any content has been copied to an email to the actual email addresses of any customers whose proxy email addresses have been found" (page 11, lines 2 and 3).

Independent Claim 2

Independent method claim 2 claims a method of proxy email address management (Fig. 9, specification p. 10, line 9 - p. 11 line 12). that includes (a) "associating a proxy email address with a customer having a customer's actual email address" (shown at 194, Fig. 8b). The method of claim 2 further includes (b) "receiving an email intended for the customer at the proxy email address" (indicated at 136 in the Fig. 9 flow chart, page 10, lines 9 and 10). Additionally, the claim 2 method includes (c) "forwarding the email to the customer's actual email address" (Fig. 9, block s148 and 150, page 10, line 32 - page 11, line 3).

Claims Dependent from Claim 2

Claims 3 - 5, 19 - 26 are dependent claims. These depend from claim 2 or from one or more claims dependent from claim 2.

Claim 3 is dependent from claim 2. This claim provides that the method of claim 2 further comprises (d) "recording the proxy email address in a database in association with the customer's actual email address" (indicated at 194 in Fig. 8b, page 9, lines 23 - 27), and (e) "looking up the customer's actual email address upon receiving the email at the proxy email address" (block 138 of Fig. 9, page 10, lines 23 - 24).

Claim 4 is dependent from claim 3. It adds the step of "causing a Web page to be displayed that bears the proxy email address" (The "WHOIS," Fig. 3, block 27, item 23, Fig. 10, page 7, lines 22 – 29, Fig. 8c, items 202 and 23, page 9, lines 27 - 31).

Claim 5 is dependent from claim 3. Claim 5 further provides the steps of "receiving a second email from the customer" (Fig. 7, items 20, 132 and 60, page 9, lines 8 - 12) and "forwarding the second email to a third party identified by the customer" (Fig. 7, items 60, 130 and 29, page 9, lines 8 - 12).

Dependent claim 19 is dependent from independent claim 2. This claim provides "receiving all incoming proxy emails to a common catch-all account" (page 10, lines 13 and 14) and "periodically checking for new emails for any customer" (page 10, lines 15 - 17).

Dependent claim 20 depends from claim 19. Claim 20 provides that "periodically checking for new emails for any customer comprises checking addresses of all addressed and copied parties for a proxy email address associated with any customer whenever a new email is found" (Page 10, lines 32 - 34).

Dependent claim 21 depends from claim 20. It calls for "forwarding the email to the customer's actual email address comprises copying the content of the email into a second email to the customer's actual email address" (page 10, lines 32 - 34).

Dependent claim 22 is dependent from claim 21 and provides "deleting an email once addresses of all addressed and copied parties have been checked for proxy email addresses of customers and the email has been copied to the actual email addresses of any customers whose proxy email addresses have been found" (page 11, lines 2 and 3).

Claim 23 is dependent from independent claim 2 and adds the step of "receiving a non-email message sent to the customer at a location based on publicly available proxy contact information" (page 11, lines 6 - 9).

Claim 24, dependent from claim 23, further provides that "receiving non-email messages comprises receiving a message by ground carrier service" (page 11, lines 6 - 9).

Claim 25 is dependent from claim 23. This claim provides "alerting the customer to the receipt of the non-email message" (Fig. 13, page 11, lines 10 and 11). It also provides "affording the customer the opportunity to request receipt of the non-email message" (Fig. 13, item 214, page 11, lines 11 and 12) and "forwarding the non-email message to the customer if the customer requests receipt of the non-email message" (page 11, lines 4 and 5).

Claim 26 is dependent from dependent claim 4. It provides that step (f) of claim 4 comprises "saving the proxy email address into whois data for a domain name" (Fig. 7, items 24, 27 and 60, page 9, lines 32 - 34).

Independent Claim 6

Independent claim 6 calls for a proxy email computer installation including "a database in computer memory storing customer information in association with a proxy email address,

wherein the customer information includes a customer's actual email address" (indicated at 194 in Fig. 8b, page 9, lines 23 - 27). The computer installation of claim 6 has "an email server to receive an email intended for a customer at the proxy email address" (Fig. 9, block 136, page 10, lines 9 - 12). The claim 6 installation includes "computer executable code on a computer-usable medium or media providing" (represented by the Fig. 9 flow chart, page 10, line 10) providing "first programming to detect the email received at the email server intended for the customer" (block 138 of Fig. 9, page 10, lines 12 - 13). The code also provides "second programming for retrieving the customer's actual email address from the database upon receipt of the email" (block 138 of Fig. 9, page 10, lines 23 - 24) and "third programming to forward the email to the customer's actual email address" (Fig. 7, items 20, 60 and 132, page 9, lines 2 - 4). The installation of claim 6 provides, as well, "a connection to a communication link for forwarding the email to the customer's actual email address" (Fig. 7, link 132, page 9, lines 2 - 4).

Claims Dependent from Claim 6

Dependent claim 7 adds to the proxy email computer installation of claim 6 "a filtering software in computer-executable code on a computer-usable medium or media for preventing an objectionable email being forwarded to the customer" (Fig. 9, blocks 144, 146, page 10, lines 27 - 32).

Claim 8 depends from claim 7 and provides "in said database, an indication of customer's choice or choices for an email filtering" (page 9, lines 5 - 7).

Claim 9, dependent from claim 8, provides for customer choices of:

(i) no filtering;
(ii) blocking all emails addressed to the proxy email address associated with the customer's information; and
(iii) blocking objectionable emails addressed to the proxy email address associated with the customer's information

(Fig. 9, decision blocks 140 and 144, page 10, lines 24 - 30).

Claim 10 is dependent from claim 9 and provides that "the blocked objectionable email is selected from the group consisting of SPAM, bulk email, advertising, pornography and code to

interfere with a computer's workings" (Fig. 9, decision blocks 140 and 144, page 10, lines 24 - 30).

Independent Claim 11

Claim 11 is independent. The claim is drawn to a computer program (the Fig. 9 flow chart) for proxy email address management (Fig. 7, block 60) that comprises computer-executable code on a computer usable medium or media (a) "first programming establishing a database for storing customer information in association with a proxy email address, wherein the customer information includes a customer's actual email address" (indicated at 194 in Fig. 8b, page 9, lines 23 - 27). The claim 11 program further includes (b) "second programming to identify a first email intended for a customer addressed to the proxy email address" (block 138 of Fig. 9, page 10, lines 12 - 13). The claim also provides (c) "third programming to retrieve the customer's actual email address" (block 138 of Fig. 9, page 10, lines 23 - 24), (d) "fourth programming to copy content of the first email to a second email" (page 10, lines 32 - 34), and (e) "fifth programming operative to send the second email to the customer's actual email address" (Fig. 9, item 150 and page 10, line 32 – page 11, line 2).

Claims Dependent from Claim 11

Dependent claim 12 is dependent from independent claim 11 and adds to the computer program of claim 11 "sixth programming to identify a third email from the customer" (Fig. 7, items 20, 132 and 60, page 9, lines 8 - 12) and "seventh programming for copying content of the third to a fourth email for the customer's intended recipient" (Fig. 7, items 60, 130 and 29, page 9, lines 8 - 12).

Dependent claim 13 depends from independent claim 11 and further comprises "sixth programming for filtering out objectionable emails" (Fig. 9, blocks 144, 146, page 10, lines 27 - 32).

Dependent claim 14 depends from claim 13 and further provides "seventh programming for receiving a customer's choice or choices of filtering and for storing the choice or choices in the database" (Fig. 12, page 9, line 5 - 7) and "eighth programming to recognize the customer's

choice or choices and filter or not filter emails to the customer based on the customer's choice or choices" (Fig. 9, blocks 144, 146, page 10 lines 24 - 32).

Grounds of Rejection to be Reviewed on Appeal (37 C.F.R. § 41.37(c) (1) (vi))

All of claims 1 - 26 stand rejected under 35 U.S.C. § 102(e) as anticipated by a published continuation-in-part U.S. patent application No. 2003/0191969A1 of Katsikas ("the Katsikas '969 published CIP" or "CIP application"). That application is attached for the Board's convenience at Appendix D.

Argument (37 C.F.R. § 41.37(c) (1) (vii))

First, because those portions of the relied-upon Katsikas '969 published CIP that are entitled to a "prior art date" early enough to qualify as prior art as to the present application do not disclose the proxy email provisions of the rejected claims of this application, the rejection of all claims as anticipated by the '969 published CIP is in error and cannot stand. Second, cited portions of the predecessor applications from which the Katsikas '969 published CIP claims priority do not supply the additional content needed to reject the claims as "anticipated." Third, even taken in their totality, irrespective of the filing date to which they are entitled or whether the content can be considered as part of the published '969 CIP, the Katsikas published CIP and its predecessor applications do not have disclosure capable of anticipating the invention as claimed.

Katsikas Briefly

The Katsikas published CIP application, parent application and provisional application relate to computer software used by a "subscriber" for eliminating unwanted "spam" email. The software features of the Katsikas provisional, parent and CIP applications changed with each filing. For each application, however, incoming email is compared to an "authorized sender list" or "ASL" to determine whether to block incoming email from reaching its intended recipient. Content that the examiner quotes from the provisional application in support of the final rejection is not present in the published CIP application on which the rejection is based. The published CIP application introduces for the first time a processor called variously a "SkProxy Processor" 202 (In Fig. 2), an "email proxy preprocessor" (paragraph 0027) or a "skproxy preprocessor"

(paragraph 0037). Prior to the filing of the CIP application, nothing described as a "proxy" had been disclosed.

The Relevant Applications and Their Effective File Dates

Filed March 31, 2003, the Katsikas '969 published CIP application is entitled to priority from parent application Serial No. 09/648,894 filed August 25, 2001 (the "parent '894 application") only for subject matter common to the parent and this continuation-in-part application. *Transco Products Inc. v. Performance Contracting, Inc.*, 38 F. 3d 551, 556-57, 32 U.S.P.Q. 2d (BNA) 1077, 1080 (Fed. Cir. 1994). The parent '894 application of Katsikas claims priority from a provisional application Serial No. 60/180,937 (the '937 provisional) filed February 8, 2000. Subject matter from the Katsikas '969 published CIP application relied upon in and essential to the rejection under 35 U.S.C. § 102(e) was introduced upon the March 31, 2003 filing of that continuation-in-part application. As to that content, the relied upon '969 published CIP is not prior art as to this application on appeal which has a priority date of August 30, 2002. Appellants' August 30, 2002 priority date is the filing date of an international (PCT) application serial No. PCT/US02/27956. This application on appeal is a divisional of the international application's U.S. National Stage Serial No. 10/624,883 (now U.S. patent No. 7,130,878). Appellants' priority and the lineage of this application is in the record at the application's Cross Reference to Related Applications as amended February 28, 2005.

The Basis of the Examiner's Rejection under 35 U.S.C. § 102(e) is in Error

In a December 13, 2007 response to an Official Action applicants pointed out that the portions of the Katsikas '969 CIP relied upon in the rejection of the claims were not entitled to a date sufficiently early to qualify as prior art. In the subsequent, final Official Action of March 20, 2008 the rejection was nevertheless maintained. There it was stated, incorrectly, that the subject matter not explicitly included in the prior PCT provisions and parent applications was present in software (called "SpamKapu software") that was said by the examiner to have been included in the provisional application. For the Board's convenience, a copy of the '937 provisional and '894 parent applications are attached at Appendices E and F. The SpamKapu software is not itself included in the Katsikas '937 provisional, but rather, aspects of that software, as of the time of filing, are described and flow-charted.

"Proxy email" installations, methods and programming are the subject of each of the independent claims 1, 2, 6 and 11 in this application. The installation, method and programming for proxy email of all claims at issue relate to the operation of an entity established to receive email addressed to a "customer" at an email address other than the customer's actual email address, in order to keep from the customer such email as the customer does not want to receive.

The Katsikas '969 published CIP application, parent '894 application, and '937 provisional relate to gate keeping software that a subscriber can use to let pass to the subscriber email from a sender whose identification appears on an "approved sender list" ("ASL"). In support of the alleged anticipation by the Katsikas published CIP in the final rejection the examiner states, at page 3 of the final Official Action:

> As per claims 1, 2, 6 and 11, Katsikas teaches a proxy email computer installation (par 0105, 0107, 0109-0110) including:
>
> A database (fig 2, elements 204, 216) in computer memory associating a customer's identification, a customer actual email and a customer's proxy email address (fig 2 and 10-11; par 0092); an email server (102) ...; a computer executable code on a computer usable medium or media providing: first programming to retrieve a customer's actual email address ... (par 0092-0094); second programming to forward (redirector 203) a second email message to the customer's actual email address; and a connection to a communication link forwarding (106) the second email message to the customer's actual email addresses (fig 1-2 and 10-11; par 0043-0045; 0092-0094 and 0100).

As for the assertion "Katsikas teaches a proxy email computer installation," the paragraphs 0105, 0107, 0109 - 0110 that the examiner cites in support are not available as prior art as to the claims of this application. Paragraphs 0105, 0107, 0109, - 0110 were new matter when the Katsikas CIP application was filed on March 31, 2003. The Katsikas '969 published CIP application attached here as Appendix D has indicated the content introduced only as of the continuation-in-part. March 31, 2003 filing date.

The Katsikas '969 published CIP on which the examiner bases the rejection does refer to, for example, "a proxy preprocessor 202," "proxy addresses," and a "Skproxy Instantiation Table" at pars. 27, 37, 91 – 102 and 100 – 102 and in Figs. 2, 10 and 12, but only in the "new matter" parts of the CIP first introduced in the application on the March 31, 2003 CIP filing date.

Concerning the individual Katsikas elements of the above-quoted statement in support of the rejection, elements 204 and 216 of Fig. 2 of the Katsikas '969 published CIP were present in

the earlier filed Katsikas applications. However, elements 204 and 216 of Fig. 2, the "Spam Processor," and the "Other Data Sources," do not related to a "database" to associate a customer's identification, actual email address and proxy email address as the examiner asserts. The function of Katsikas' "spam processor" is at paragraph 37 of the '969 published CIP.

> The Redirector 203 sends a request for validation for the email from the Spam Processor 204 which maintains the Spam Processing Database (SPDB) 205, including the Authorized Senders Rules List (ASL) 206. The SPDB Database and ASL Rules List are the heart of SPAMKAPU, as they contain the processing rules and lists of persons authorized to send email to the respective SUBSCRIBERS of the system. The Spam Processor 204 sends a response, either that the sender's address on the email is not authorized on the ASL List, i.e., is a SPAMMER, or is authorized on the ASL rules list, i.e., is a FRIEND, or is not present at all on the ASL rules list, i.e. is a UNKNOWN.

In other words the "Spam processor 204" is involved in communications from and to email senders who are possible spammers, not the subscribers (or "customers" as the claims put it) that are to be protected from spam.

Similarly the element 216 of the Katsikas '969 published CIP that the examiner cites is described as containing data such as "header data from sent email and data from other data sources," (paragraph 0042). It was never, as is contended in the examiner's rejection, said to be "associating a customer's identification, a customer actual email, and a customer's proxy email addresses." The examiner's mistaken characterization of the Katsikas '969 published CIP's content in this respect is a central flaw in the anticipation rejection of every claim in this application because each independent claim in the application includes claim content associating the proxy and "actual" email addresses of a customer.

The further application figures, Figs. "10-11," and the paragraph, "par 0092," that the examiner cites as to the "database" were new matter in the Katsikas '969 published CIP upon filing. They are not entitled to the filing date of either of the earlier Katsikas '894 parent or '937 provisional application.

In the above-quoted segment of the examiner's rejection the examiner correctly cites element 102 of Katsikas as an "email server." This does not alter the failure of the Katsikas published CIP to anticipate the claims however.

Paragraphs 0092 - 0094 are cited in the foregoing quote from the final rejection for teaching the claimed "code ... providing ... first programming to retrieve a customer's actual email address." But paragraphs 0092 to 0094 are new matter as to the Katsikas '969 CIP and unavailable to support the rejection. (That the cited paragraphs do not actually provide what is stated in the final Official Action is treated below.)

"Figs. 1 and 2 and 10, 11" and paragraphs 0043 to 0045 and 0092 to 0094 and 0100 the examiner cites for a communication link (identified as 106) for sending a second email "to the customer's actual email address." Figs. 1 and 2 of the Katsikas '969 published CIP show nothing of the kind. Item 106 is a generalized indication of the internet in Figs. 1 and 2. In Figs. 1a and 1b email is shown received from the internet 106. This is email for a subscriber or customer ('969 published CIP paragraph 0028 as to Fig. 2a, paragraph 0030 as to Fig. 1b). Mail is not shown being sent through the internet 106 to the subscriber or customer as contended by the examiner. Rather, the email shown being sent via the "customer email client 101," which is the internet service provider such as Outlook or Netscape, is being sent <u>from</u>, not <u>to</u>, the subscriber or customer (again, paragraphs 0028, 0030 of the '969 published CIP application). The Katsikas '969 CIP Fig. 2 shows subscriber or customer email client 101 sending the customer's email message via a SMTP send manager 214 and standard "SMTP Send process." There is no suggestion in any of this of sending a second email message to a customer's actual email address as alleged in the paragraph quoted above.

Given the foregoing then, the rejection entirely misstates the operations described in the published CIP and by virtue of that the final rejection in this application is in error and should be overturned even before considering what if anything further from the preceding '894 parent and '937 provisional can be relied upon to supplement the teaching of the '969 published CIP.

The Rejection's Flawed Grounds for Relying on Non-Prior Art Subject Matter in the Continuation-in-Part

To justify persisting in the original rejection of claims in this application as anticipated by the Katsikas '969 published CIP application, the examiner contends that the '937 provisional patent application from which the Katsikas '969 published CIP application claims priority contains subject matter that supports the new matter that was introduced into or teaches what is lacking from the relied upon CIP. This basis for maintaining the mistaken rejection over the published CIP errs in three ways.

First, the '937 provisional application content that the examiner cites to justify the rejection based on of the Katsikas '969 published CIP was dropped from the Katsikas line of applications and never became a part of the published application entitled to the '937 provisional application filing date. Second, that part of the earlier '937 provisional application that the examiner cites does not actually teach, support, expand on or explain the relied-upon new matter provisions or needed or missing features of the '969 continuation-in-part published application and so cannot support the rejection. Third, and most important, even if the content of the provisional application referred to by the examiner were considered to be a part of the '969 Katsikas published CIP application, the application still would not meet the terms of the rejected claims.

At page 4 of the final Official Action, the examiner states in the Response to Arguments in the final rejection:

> Applicants argued that Katsikas does not qualify as prior art for this application because the provisional application "937" and the parent application "894" lack pertinent details such as proxy email as recited by Katsikas "969."
>
> Examiner submits that although Katsikas does not elaborate on email proxy, this feature is included in SpamKapu software (see pages 7-8 of "937"; fig. 1-8).

The examiner then quotes from the '937 provisional application:

> As a server-side software package or online service SUBSCRIBERs are added to SpamKapu system. Each SUBSCRIBER is provided with a PSM, ASM, and UMM and an SKE.

> The SUBSCRIBER changes appropriate setting on their email software to accomplish the following:
>
> Use current interact standards (currently POP3 or IMAP4) to retrieve mail from both the PSM and ASM.
>
> Redirect email sent to their current email address to their SKE instead OR set the email reply-to address to their SKE. Use the SMTP manager to handle the sending of all email. Any email sent to the SKE is processed by the redirector as described above. Any email sent by the SUBSCRIBER through the ASL manager (via the SMTP manager) and processed as described above.
>
> The user can retrieve email from the ASM at any time using Internet standards (currently POP3 or IMAP4). The user can retrieve email from the PSM at any time using internet standards (currently POP3 or IMAP4) user other software that can delete, further filter, or altogether discard the contents.
>
> SUBSCRIBERs may interact with the UMM at any time.

The above passage and other text of the '937 provisional uses acronyms unique to that application and others known in the art. POP3 and IMAP4 are used in the art for known protocols. ASM is used to mean "authorized sender mailbox," UMM is "user maintenance module," SKE is "Spamkapu email address" and ASL is "authorized senders list." The definition of PSM was not found.

At page 5 of the final Official Action the examiner concludes:

> Examiner concludes that although some of the contents of the provisional application and the CIP are different, the features and elements needed to reject the present application are similar in both references. Therefore, Katsikas qualifies as prior art to reject the claims of the instant application. Accordingly, the rejection is maintained.

Regarding availability as prior art of the Katsikas provisional application content that was not carried forward into the Katsikas publication, 35 U.S.C. § 102(e)(1) provides that "a person shall be entitled to a patent "unless -...(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the application for patent,-...."

The provisional application contents relied upon by the examiner in support of the Section 102(e)(1) rejection of the claims in the application were not "described in an application for patent, published under Section 122(b) ...." Those contents did not become a part of the

subsequently published '969 application. For that reasons it is submitted the '937 provisional application section relied upon by the examiner is not entitled to the "prior art date" afforded by 102(e)(1) and does not therefore qualify to support the Section 102(e)(1) rejection by filling in crucial missing content of the Katsikas '969 published CIP. It is noted in passing that the Katsikas '969 published CIP application does not claim to incorporate the '937 provisional or the parent '894 application by reference. Further, nothing in 35 U.S.C. § 122(b) relating to publication of a pending application suggests that a preceding and unpublished provisional application's content should be considered to constitute any part of the application ultimately published under the provisions of that section. And Section 122(b)(2)(iii) indicates that a provisional application is not to be published under the publication provisions of 122(b).

In the present case there is left as available prior art to support the section 102(e) rejection only that published content of the Katsikas '969 published CIP that was not new matter upon filing of the CIP. As established above, and apparently acknowledged by the examiner, that published CIP content does not meet the terms of the claims in this application.

Even Combined, the Several Katsikas Applications Do Not Anticipate the Invention Claimed

The above-noted second and third bases for error in the outstanding rejection are related. The provisional application content that the examiner cites does not relate to a proxy email installation method or programming such that it either (1) overcomes the published '969 continuation-in-part application's failure in this respect or (2) teaches the features of the claimed invention such that an anticipation rejection would be proper if the provisional application's content were, in fact, prior art as respects the present application.

It is well established that anticipation of a patent claim by a prior art reference requires that reference to "read on each limitation of the claim." *Verdegual Bros. v. Union Oil Co. of Calif.*, 814F.2d 628, 631; 2USPQ2d 1051, 1053 (Fed. Cir. 1987). The Katsikas '969 published continuation-in-part application does not meet the terms of the claims of this application as the examiner acknowledges. However, neither do the contents of the '937 provisional upon which the examiner relies.

Each of the independent claims 1, 2, 6 and 11 sets forth an installation, method or program that includes or deals with a customer's "actual email address," and "a proxy email address." Neither the Katsikas provisional application nor the Katsikas parent application describes or utilizes two differing email addresses that can be considered to be an actual and a proxy email address. In each of the claims in this application, the proxy email address and the customer's actual email address are used in a fashion not contemplated by any of the Katsikas applications. Consider each independent claim in the application. Claim 1 calls for "a database in computer memory associating a customer's identification, a customer's actual email address and a customer's proxy email address." Claim 2 calls for "associating a proxy email address with a customer having a customer's actual email address." Claim 6 calls for "a database in computer memory storing customer information in association with a proxy email address, wherein the customer information includes a customer's actual email address." And claim 11 calls for "first programming establishing a database for storing customer information in association with a proxy email address, wherein the customer information includes a customer's actual email address." These provisions and their association are not found in the Katsikas '969 published CIP, '984 parent application or '937 provisional application. Consequently it cannot be said that the '937 provisional application overcomes the deficit of the Katsikas '969 published CIP in this respect. And it cannot be said that combined the Katsikas published CIP, parent and provisional applications have what is necessary to anticipate the independent claims 1, 2, 6 and 11.

As for dependent claims in the application they patentably differ from the Katsikas applications on the basis of their dependencies, but they do not stand or fall with the independent parent claims 1, 2, 6 and 11.

Nowhere in the Katsikas applications are the recited features of claim 3 - the features of:

> (d) recording the proxy email address in a database in association with the customer's actual email address; and
>
> (e) looking up the customer's actual email address upon receiving the email at the proxy email address.

The examiner states at page 4 of the final Official Action:

> As per claims 3-5, Katsikas teaches recording proxy email address in a database ...; looking up customer's actual email address; causing a web page to be displayed; receiving a second email from the customer; and forwarding the second to a third party identified by the customer (fig 1-2 and 10-11; par 0043-0045; 0092-0094 and 0100).

Again as pointed out above, none of the Katsikas applications teach recording proxy email addresses in association with actual email addresses. And the cited provisions of the published '969 CIP application, even the new matter provisions thereof relied on by the examiner, do not describe that or the looking up of the actual email address upon receiving email at a proxy email address.

Likewise the Katsikas applications do not show "causing a Web page to be displayed that bears the proxy email address" as called for in claim 4 or "forwarding" a "second email" (from the customer) to a "third party identified by the customer." See the preceding discussion of the receiving and sending of email via the internet 106 in the Katsikas application.

Regarding the dependent claims 7 – 10, as discussed above, the Katsikas applications describe preventing email reaching a subscriber unless the sender is on an authorized sender's list. Contrary to the examiner's further assertion at page 4 of the final Official Action regarding claims 7 – 10, 13 and 14, the Katsikas applications do not speak of "filtering software" (claim 7), a database "indication of customer's choice" in filtering (claim 8), the three available choices of "no filtering," "blocking all emails," and "blocking objectionable emails" (claim 9), or the filtering choices of claim 10.

With respect to the claims depending from claim 11, claim 12's programming to identify a third email from the customer and for copying its content to a fourth email to the customer's intended recipient is unlike anything in the email receiving and sending of the Katsikas applications. Pertinent here again is the discussion above respecting the error in the rejection of the independent claims in relation to email sent and received through the internet 106. There, the programming of claim 12 relating third and fourth emails to and from the customer is not to be found in the Katsikas applications.

As to claims 13 and 14, the preceding comments relating to claims 7 and 8 regarding "filtering" and "filtering choices" are pertinent.

Dependent claim 15 sets forth additional details of the programming of the appellants' software handling email for customers. The claim provides that "the email server is set up to receive all incoming proxy emails to a common catch-all account, and the email computer installation further comprises an email address management system that periodically checks for new emails for any customer." No such feature is taught in any of the Katsikas applications.

Dependent claims 16 - 18 include by their dependency the provisions of claim 15 and like claim 15 are not anticipated by any of the Katsikas applications' teachings. These are patentable over the Katsikas published application and its predecessors for this reason as well as for the particular additional terms of these claims.

Claim 19 is similar to claim 15, but dependent from claim 2. The claim patentably differs from the Katsikas applications for the reasons set forth above with respect to both claim 2 and claim 15.

Claim 20 is dependent from claim 19. It is similar to dependent claim 16 and patentable over the Katsikas applications' content for the reasons set forth above with respect to parent claims 2 and 19 and claim 16.

Claim 21 depends from claim 20, has content similar to claim 17 and is patentable over the Katsikas application for the reasons stated in regard to claims 2, 20 and 17.

Claim 22 depends form claim 21. It contains subject matter similar to claim 18 and patentably differs from the Katsikas application as discussed above with respect to claims 2, 21 and 18.

Claim 23 depends from claim 22. It patentably differs from the Katsikas applications as discussed respecting claims 2 and 22 and further calls for "receiving a non-email message sent to the customer at a location based on publicly available proxy contact information." This is not taught by Katsikas.

Claims 24 and 25 are dependent from claim 23 and patentable over Katsikas for all of the same reasons as claim 23. Additionally, unlike Katsikas, claim 24 calls for "receiving non-email messages comprises receiving a message by ground carrier service," and claim 25 calls for "alerting the customer to the receipt of the non-email message, affording the customer the opportunity to request receipt of the non-email message, and forwarding the non-email message to the customer if the customer requests receipt of the non-email message." Claims 24 and 25 patentably differ from the Katsikas applications by these recitations.

Claim 26 is dependent from claim 4. It additionally recites "step (f) [of claim 26] comprises saving the proxy email address into whois data for a domain name." The claim is patentable by its dependency from claim 4 and by its additional content absent from the Katsikas application.

Conclusion

For each of the above reasons, the final rejection of all of claims 1 – 26 as anticipated by the Katsikas '969 continuation-in-part application under 35 U.S.C. § 102(e)(1) is in error and should be reversed. Each of claims 1 – 26 should be held patentable and allowed at this time.

Respectfully submitted,

GALLAGHER & KENNEDY

Date: December 5, 2008

By: Thomas D. MacBlain
Reg. No. 24,583
Attorney for Appellant

Gallagher & Kennedy, P.A.
2575 East Camelback Road
Phoenix, AZ 85016-9225
602-530-8088
tdm@gknet.com

APPENDIX A

Claims, 37 C.F.R. § 41.37(c) (1) (viii)

1. A proxy email computer installation including:

(a) a database in computer memory associating a customer's identification, a customer's actual email address and a customer's proxy email address;

(b) an email server having an ability to receive a first email message at the customer's proxy email address;

(c) computer executable code on a computer usable medium or media providing:

(i) first programming to retrieve a customer's actual email address from the database upon receipt of the first email message to the customer's proxy email address;

(ii) second programming to forward a second email message to the customer's actual email address, wherein the second email message is formed from the first email message; and

(d) a connection to a communication link forwarding the second email message to the customer's actual email addresses.

2. A method of proxy email address management including:

(a) associating a proxy email address with a customer having a customer's actual email address;

(b) receiving an email intended for the customer at the proxy email address; and

(c) forwarding the email to the customer's actual email address.

3. The method according to claim 2, further comprising:

(d recording the proxy email address in a database in association with the customer's actual email address; and

(e) looking up the customer's actual email address upon receiving the email at the proxy email address.

4. The method according to claim 3, further comprising:

(f) causing a Web page to be displayed that bears the proxy email address.

5. The method according to claim 3, further comprising:

(f) receiving a second email from the customer; and

(g) forwarding the second email to a third party identified by the customer.

6. A proxy email computer installation including:

(a) a database in computer memory storing customer information in association with a proxy email address, wherein the customer information includes a customer's actual email address;

(b) an email server to receive an email intended for a customer at the proxy email address;

(c) computer executable code on a computer-usable medium or media providing:

(i) first programming to detect the email received at the email server intended for the customer;

(ii) second programming for retrieving the customer's actual email address from the database upon receipt of the email;

(iii) third programming to forward the email to the customer's actual email address; and

(d) a connection to a communication link for forwarding the email to the customer's actual email address.

7. The proxy email computer installation according to claim 6, further comprising:

(e) a filtering software in computer-executable code on a computer-usable medium or media for preventing an objectionable email being forwarded to the customer.

8. The proxy email computer installation according to claim 7, further comprising, in said database, an indication of customer's choice or choices for an email filtering.

9. The proxy email computer installation according to claim 8, wherein the customer's choice or choices includes one of:

(i) no filtering;

(ii) blocking all emails addressed to the proxy email address associated with the customer's information; and

(iii) blocking objectionable emails addressed to the proxy email address associated with the customer's information.

10. The proxy email computer installation according to claim 9, wherein the blocked objectionable email is selected from the group consisting of SPAM, bulk email, advertising, pornography and code to interfere with a computer's workings.

11. A computer program for proxy email address management comprising in computer-executable code on a computer usable medium or media:

(a) first programming establishing a database for storing customer information in association with a proxy email address, wherein the customer information includes a customer's actual email address;

(b) second programming to identify a first email intended for a customer addressed to the proxy email address;

(c) third programming to retrieve the customer's actual email address;

(d) fourth programming to copy content of the first email to a second email; and

(e) fifth programming operative to send the second email to the customer's actual email address.

12. The computer program in a computer-executable code on a computer usable medium or media according to claim 11, further comprising sixth programming to identify a third email from the customer, and seventh programming for copying content of the third to a fourth email for the customer's intended recipient.

13. The computer program in a computer-executable code on a computer usable medium or media according to claim 11, further comprising sixth programming for filtering out objectionable emails.

14. The computer program in a computer-executable code on a computer usable medium or media according to claim 13, further comprising seventh programming for receiving a customer's choice or choices of filtering and for storing the choice or choices in the database, and eighth programming to recognize the customer's choice or choices and filter or not filter emails to the customer based on the customer's choice or choices.

15. The proxy email computer installation according to claim 1, wherein the email server is set up to receive all incoming proxy emails to a common catch-all account, and the email computer installation further comprises an email address management system that periodically checks for new emails for any customer.

16. The proxy email computer installation according to claim 15, wherein the first programming to retrieve a customer's actual email address includes programming to check email addresses of all addressed and copied parties for a proxy email address being served by the proxy email server whenever a new email is found by the email address management system.

17. The proxy email computer installation according to claim 16, wherein the second programming comprises programming to copy the content of the first email message into the second email message, the second email message comprising an email message from the proxy computer installation to the customer at the customer's actual email address.

18. The proxy email computer installation according to claim 17, further comprising programming in computer executable code on a computer usable medium or media to effect deletion of an email once addresses of all addressed and copied parties have been checked for proxy email addresses of customers and any content has been copied to an email to the actual email addresses of any customers whose proxy email addresses have been found.

19. The method of proxy email address management according to claim 2, further comprising receiving all incoming proxy emails to a common catch-all account, and periodically checking for new emails for any customer.

20. The method of proxy email address management according to claim 19, wherein periodically checking for new emails for any customer comprises checking addresses of all

addressed and copied parties for a proxy email address associated with any customer whenever a new email is found.

21. The method of proxy email address management according to claim 20, wherein forwarding the email to the customer's actual email address comprises copying the content of the email into a second email to the customer's actual email address.

22. The method of proxy email address management according to claim 21, further comprising deleting an email once addresses of all addressed and copied parties have been checked for proxy email addresses of customers and the email has been copied to the actual email addresses of any customers whose proxy email addresses have been found.

23. The method of proxy email address management according to claim 2, further comprising receiving a non-email message sent to the customer at a location based on publicly available proxy contact information.

24. The method of proxy email address management according to claim 23, wherein receiving non-email messages comprises receiving a message by ground carrier service.

25. The method of proxy email address management according to claim 23, further comprising alerting the customer to the receipt of the non-email message, affording the customer the opportunity to request receipt of the non-email message, and forwarding the non-email message to the customer if the customer requests receipt of the non-email message.

26. The method according to claim 4, wherein step (f) comprises saving the proxy email address into whois data for a domain name.

APPENDIX B

Evidence, 37 C.F.R. § 41.37(c) (1) (ix)

No evidence was submitted pursuant to 37 C.F.R. § 1.130, 1.131 or 1.132 nor any other evidence entered by the examiner and relied upon by appellant in the appeal.

APPENDIX C

Related Proceedings, 37 C.F.R. § 41.37 (c) (1) (x)

There are no decisions rendered by a court or the Board in any proceeding.

US 20030191969A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2003/0191969 A1

Katsikas (43) Pub. Date: Oct. 9, 2003

(54) SYSTEM FOR ELIMINATING UNAUTHORIZED ELECTRONIC MAIL

(76) Inventor: Peter L. Katsikas, Honolulu, HI (US)

Correspondence Address:
LEIGHTON K. CHONG
OSTRAGER CHONG & FLAHERTY
(HAWAII)
841 BISHOP STREET, SUITE 1200
HONOLULU, HI 96813 (US)

(21) Appl. No.: 10/404,631

(22) Filed: Mar. 31, 2003

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/648,894, filed on Aug. 25, 2000.

(60) Provisional application No. 60/180,937, filed on Feb. 8, 2000.

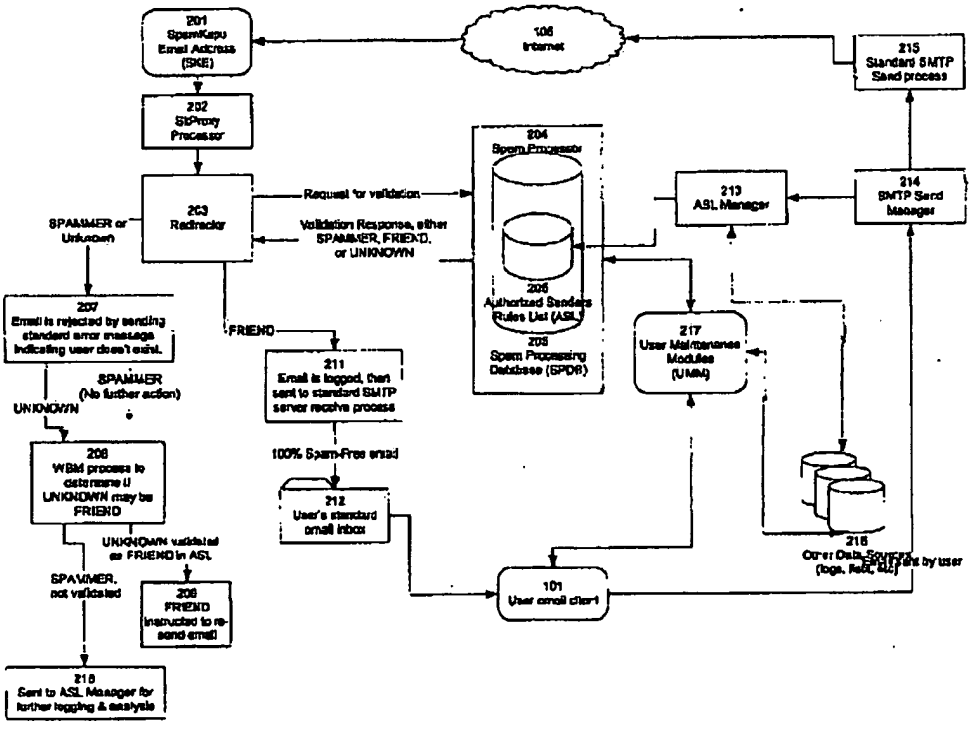
Publication Classification

(51) Int. Cl.[7] G06F 11/30; G06F 15/173

(52) U.S. Cl. 713/201; 709/224

(57) ABSTRACT

A system for eliminating unauthorized email sent to a user on a network analyzes the sender address of incoming email and determines whether it is to be rejected by returning a standard "no such user" error code or accepted depending upon executing processing rules and analyzing managed lists of authorized senders. This provides an advantage over existing anti-spam filtering systems by intercepting unauthorized email before it reaches an existing email server or client. The system rejects all email unless authorized by using a standard "no such user" error code, and by redirecting the unauthorized email back to the sender or to a sender evaluation site. An ASL module captures authorized sender addresses from the user's outgoing email and other sources in order to update "authorized senders" lists. The system may employ a WBM procedure that notifies senders of rejected email to go to a separate website and register as valid senders after passing an interaction test that precludes automatic registration by a mechanical program. A destination proxy email address procedure allows subscribers to use temporary proxy addresses for receiving email expected from unknown sources and instantiates senders as authorized upon receiving the expected email to the proxy addresses. The unauthorized-email rejection component can be readily configured as a hardware or software appliance used in tandem with a conventional email server, email gateway, or firewall to an intranet, or as a software extension to an existing firewall system.



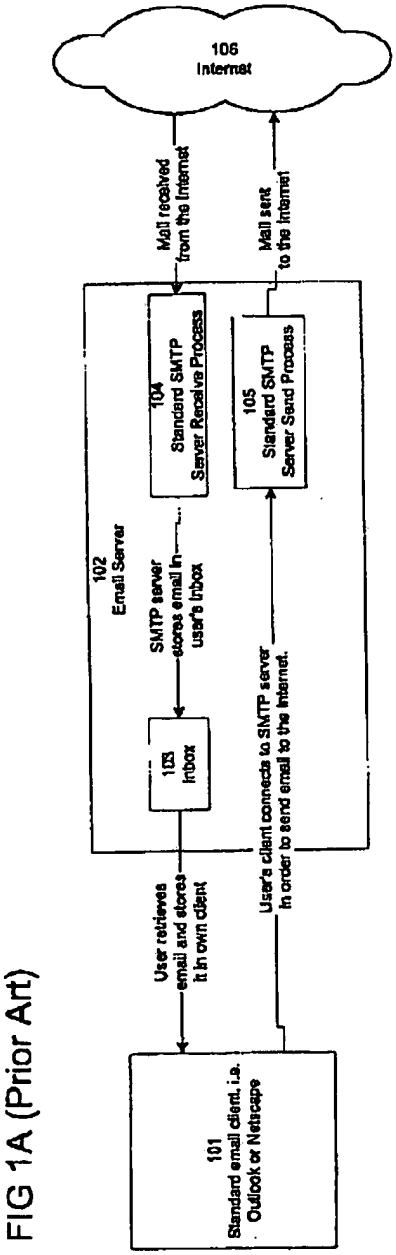


FIG 1A (Prior Art)

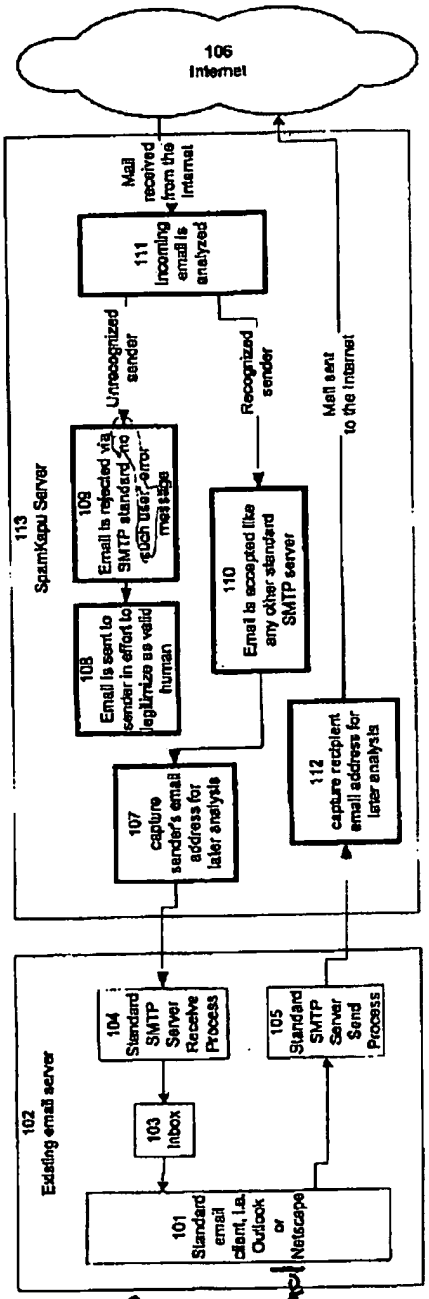


FIG. 1B

Revised from Fig. 1B of Parent '894 and '937 provisionals

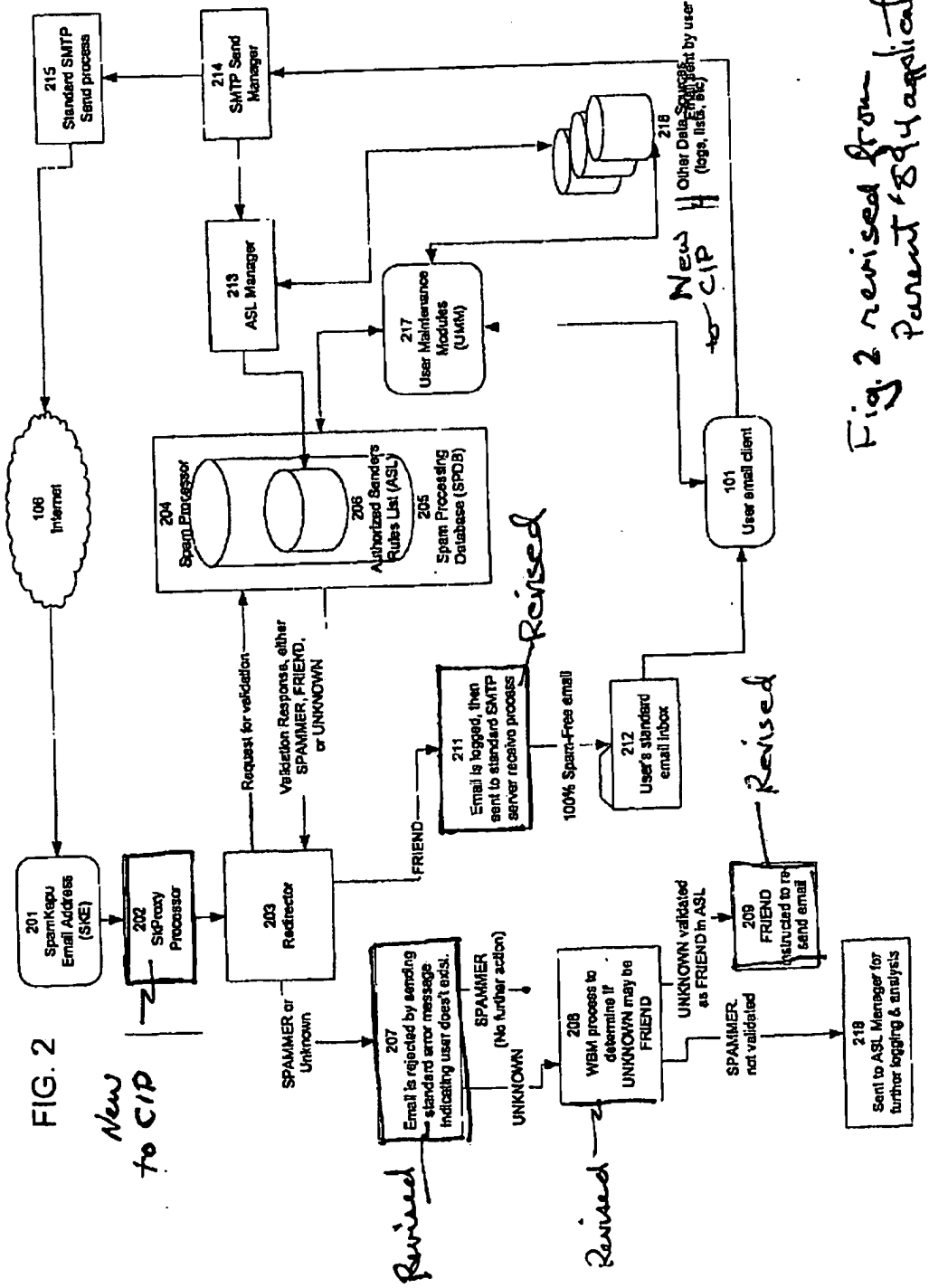


FIG. 2

New to CIP

Fig. 2 revised from parent '894 application

FIG 3a (Prior Art):

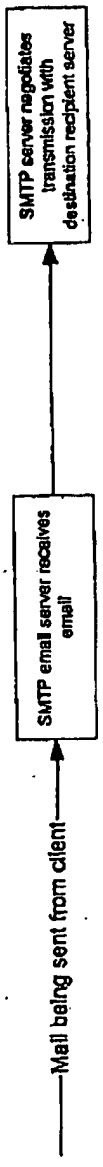
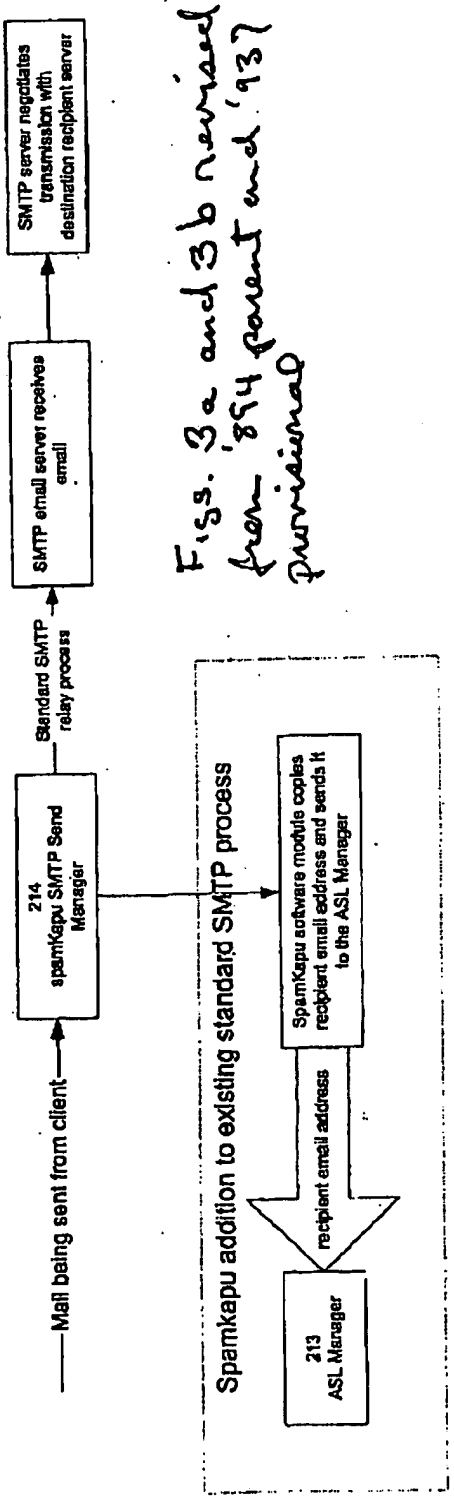


FIG 3b:



Figs. 3a and 3b removed from '854 parent and '937 provisional

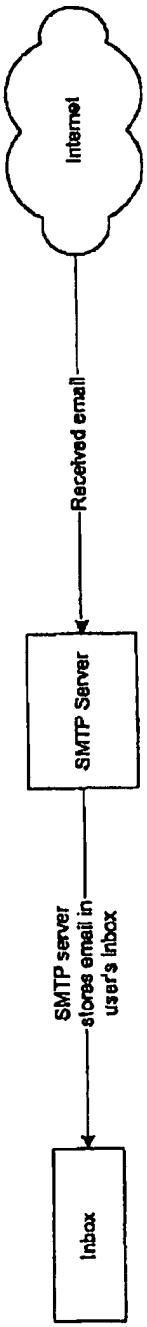


Figure 4A (Prior Art)

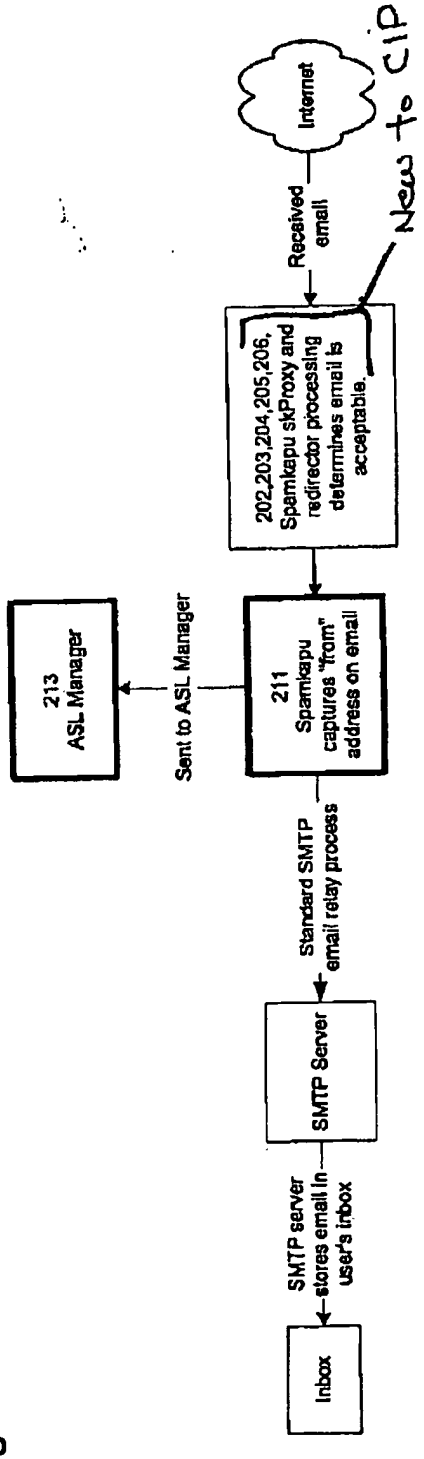


Figure 4B

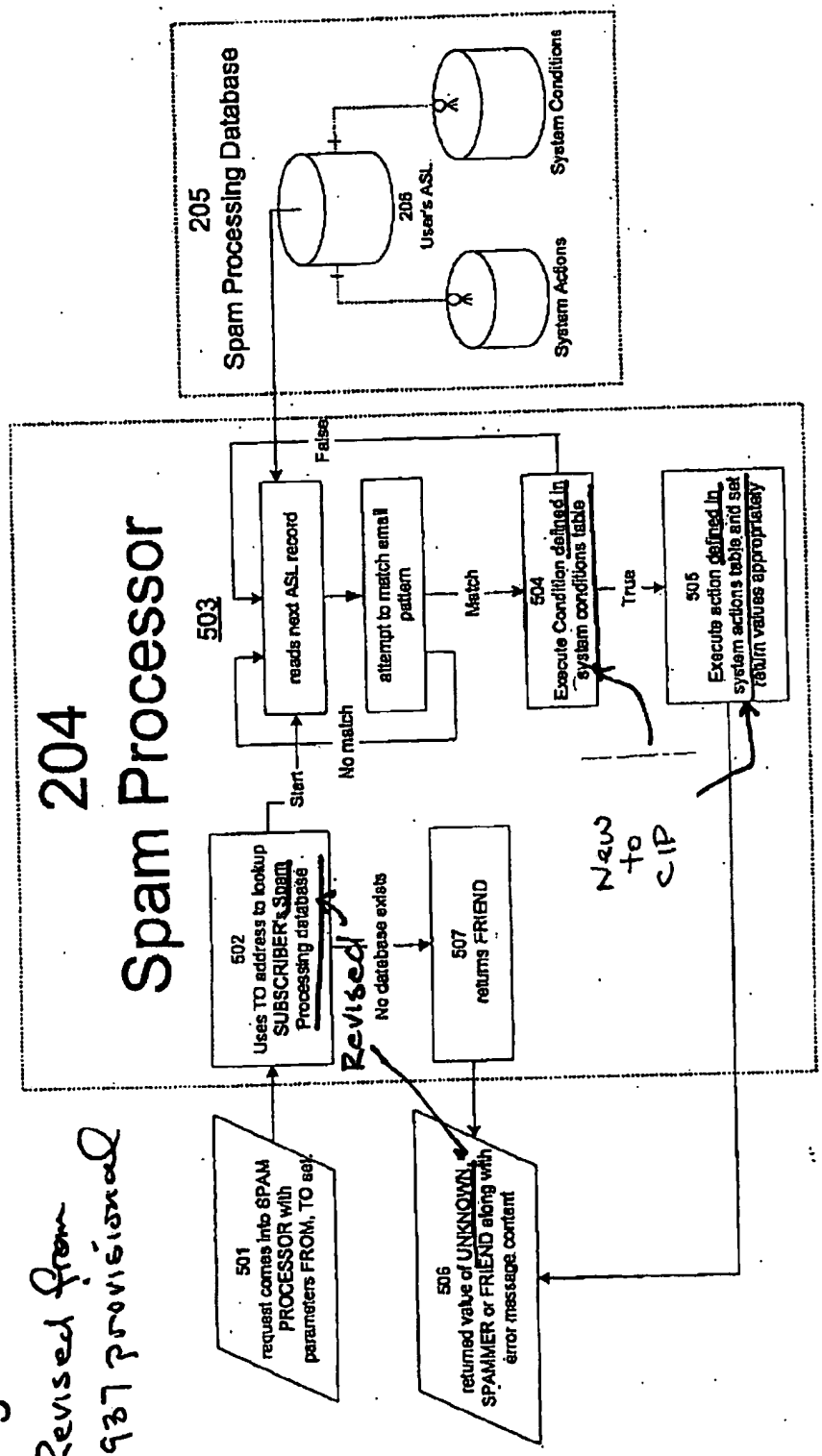


Figure 5:

Revised from 1937 provisional

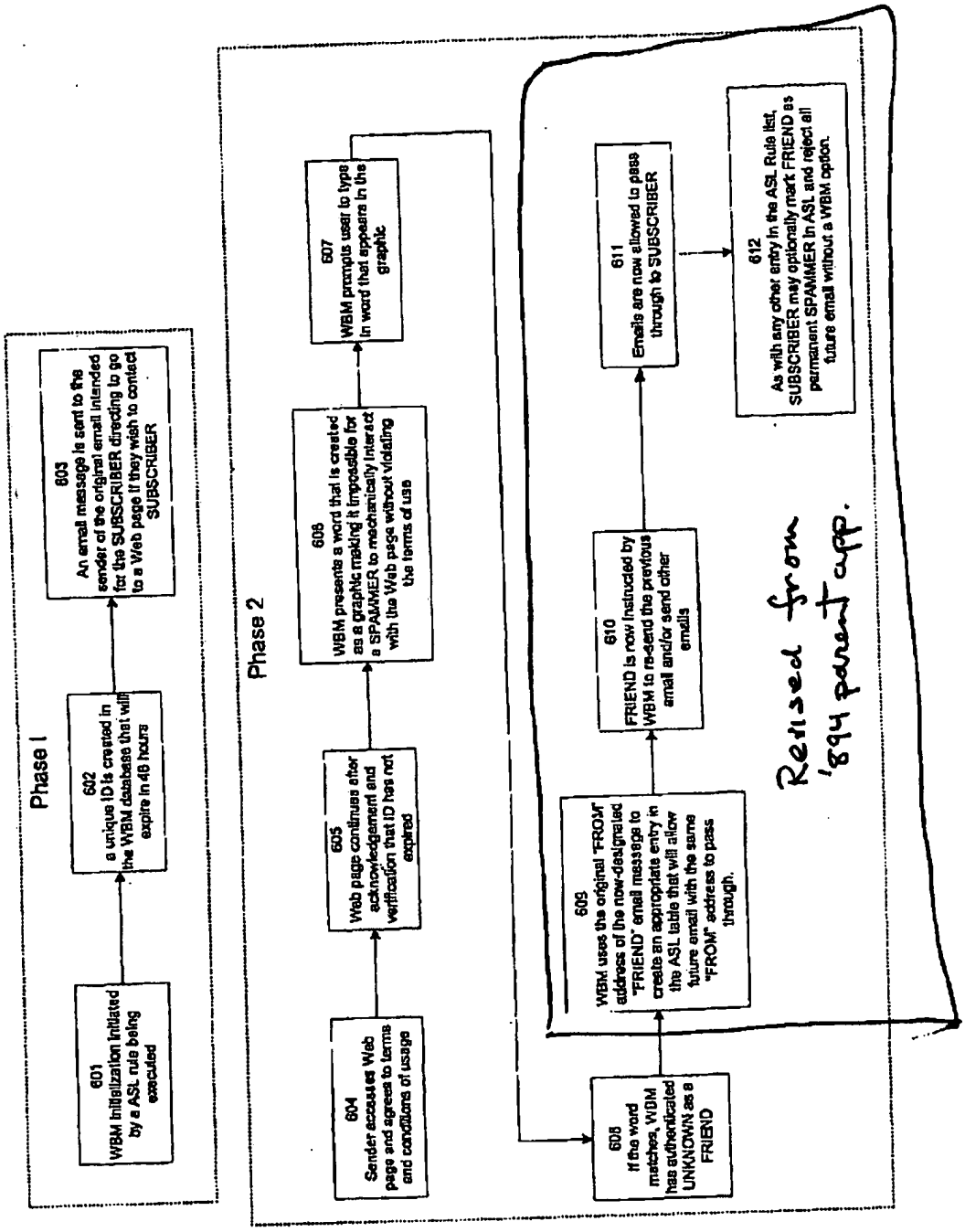


FIG. 6: Not present in Provisional

Revised from '894 parent app.

FIG 7A

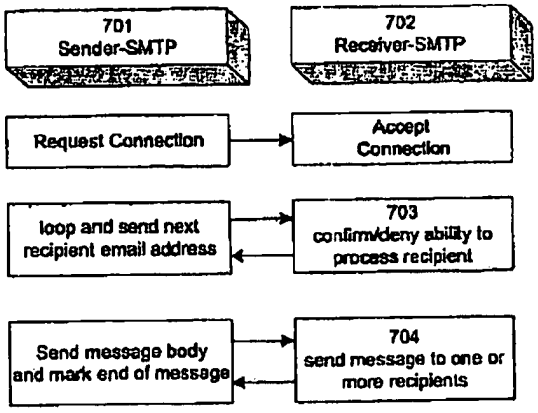


FIG 7B: Revised from '594 parent app.

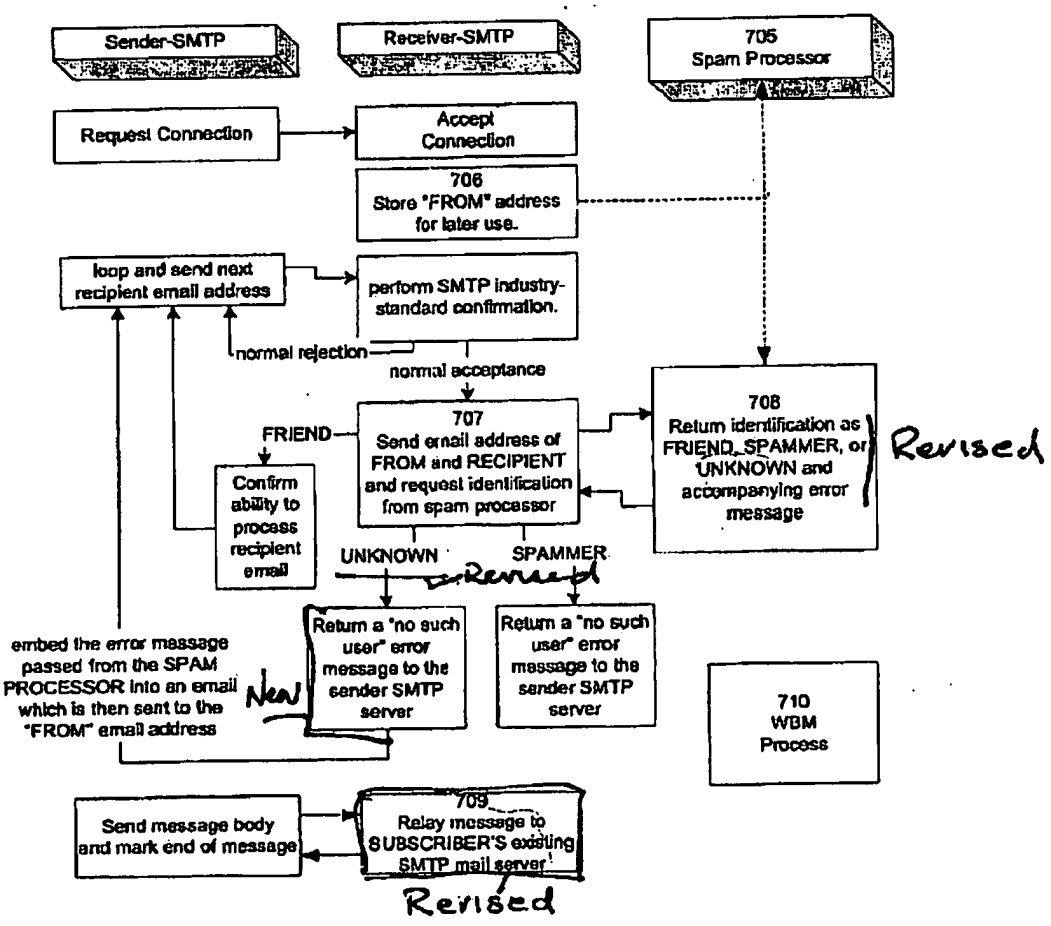
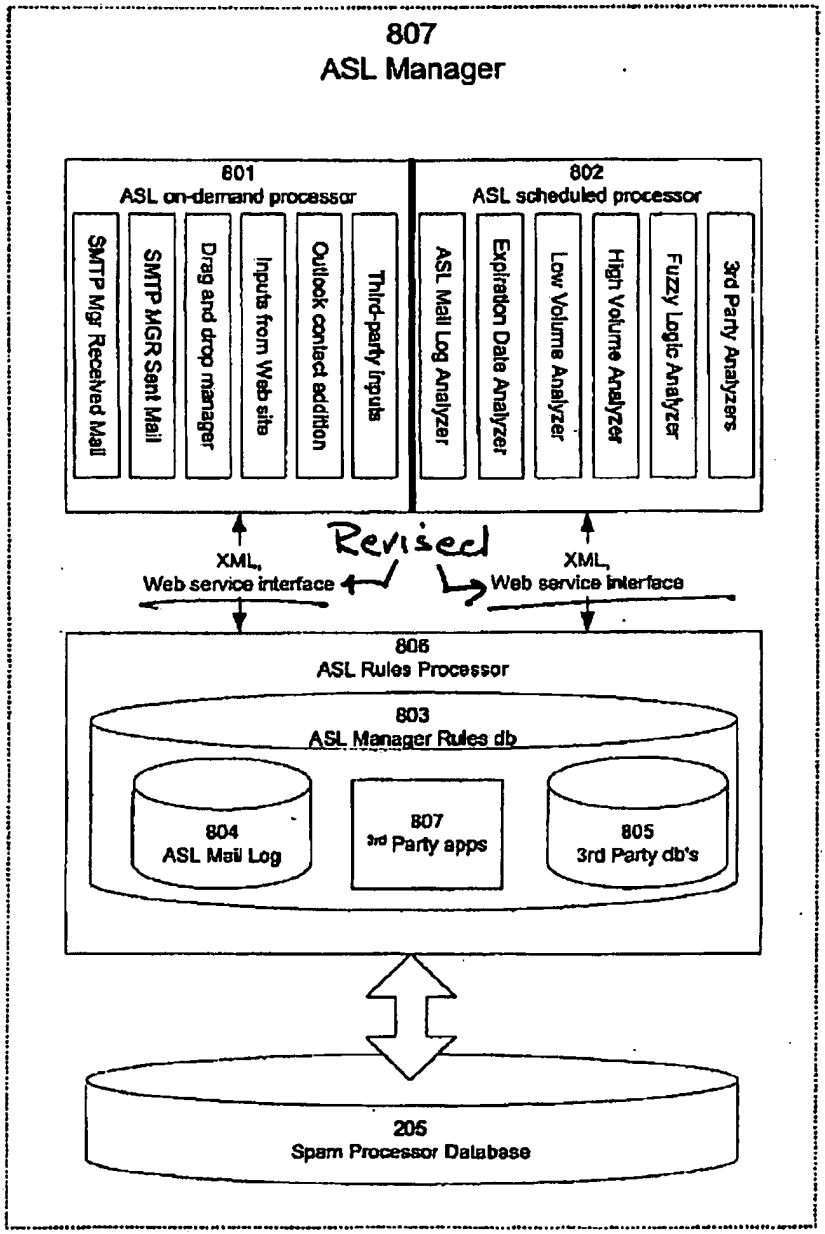


FIG 8A


807
ASL Manager
801
ASL on-demand processor
SMTP Mgr Received Mail
SMTP MGR Sent Mail
Drag and drop manager
Inputs from Web site
Outlook contact addition
Third-party inputs
802
ASL scheduled processor
ASL Mail Log Analyzer
Expiration Date Analyzer
Low Volume Analyzer
High Volume Analyzer
Fuzzy Logic Analyzer
3rd Party Analyzers
Revised
XML,
Web service interface
XML,
Web service interface
806
ASL Rules Processor
803
ASL Manager Rules db
804
ASL Mail Log
807
3rd Party apps
805
3rd Party db's
205
Spam Processor Database

All new to CIP

FIG 8B

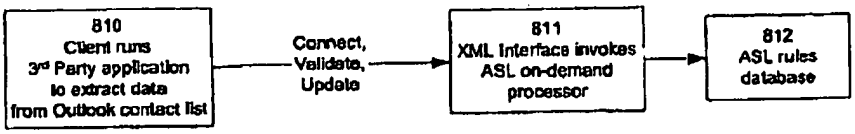


FIG 8C

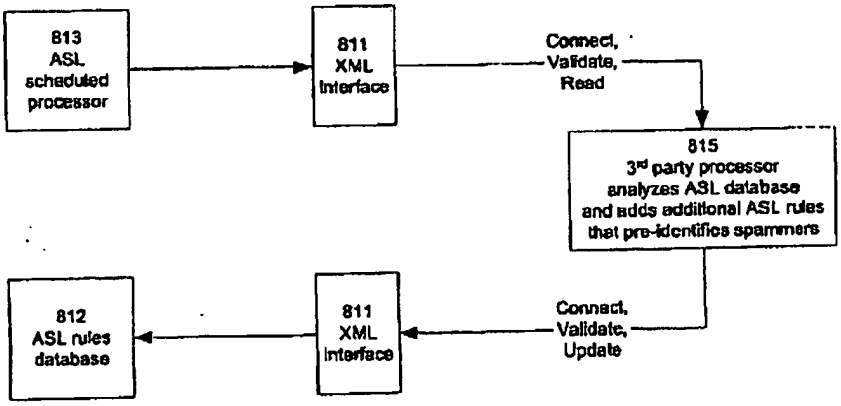


FIG 8D

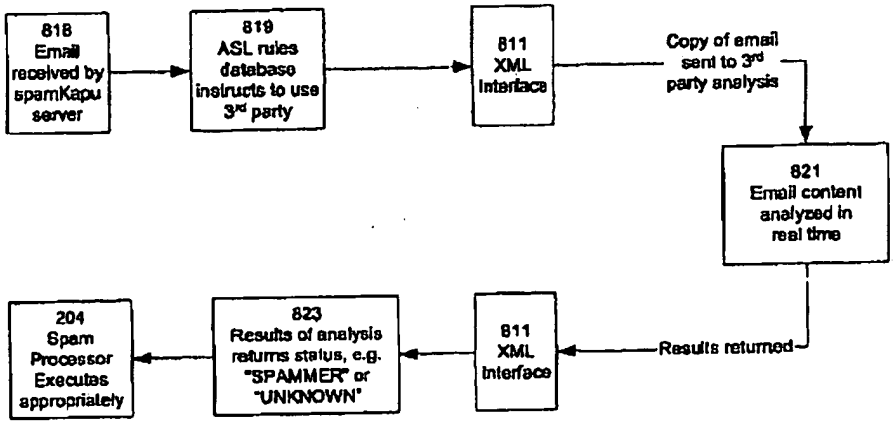
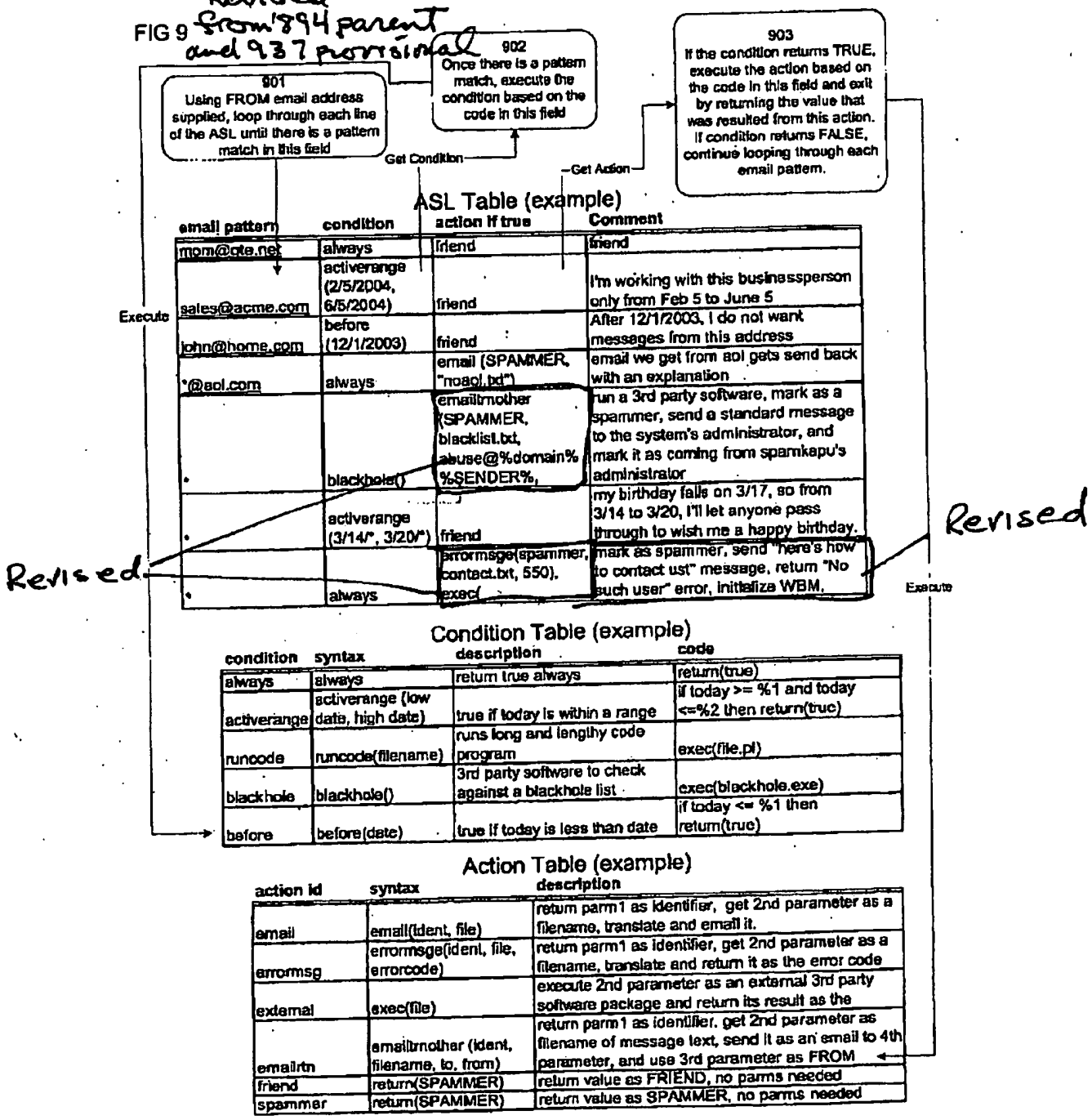


FIG 9 Revised from '894 parent and 937 provisional



All new to CIP

FIG 10A:

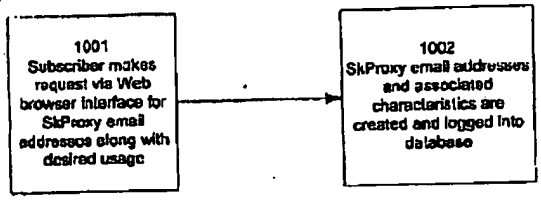


FIG 10B:

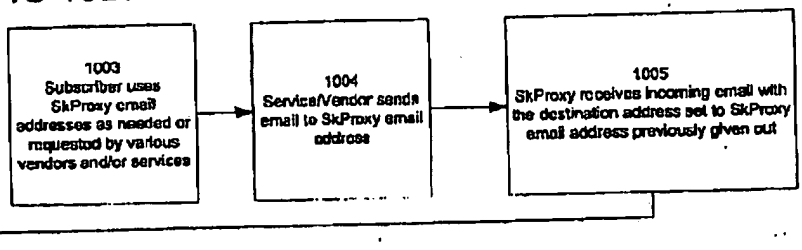


FIG 10C:

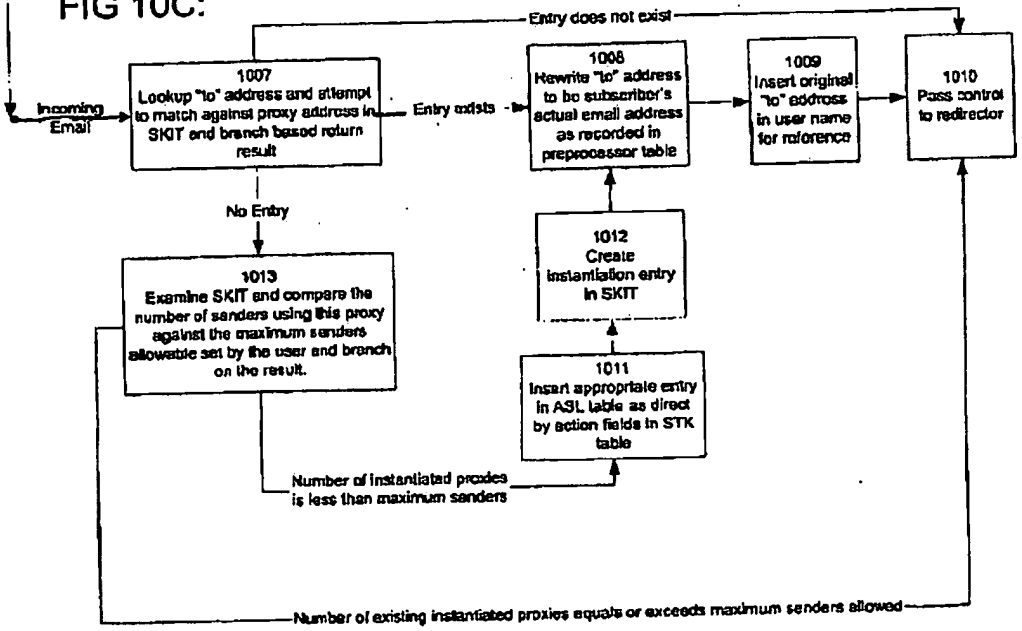


Figure 11A:

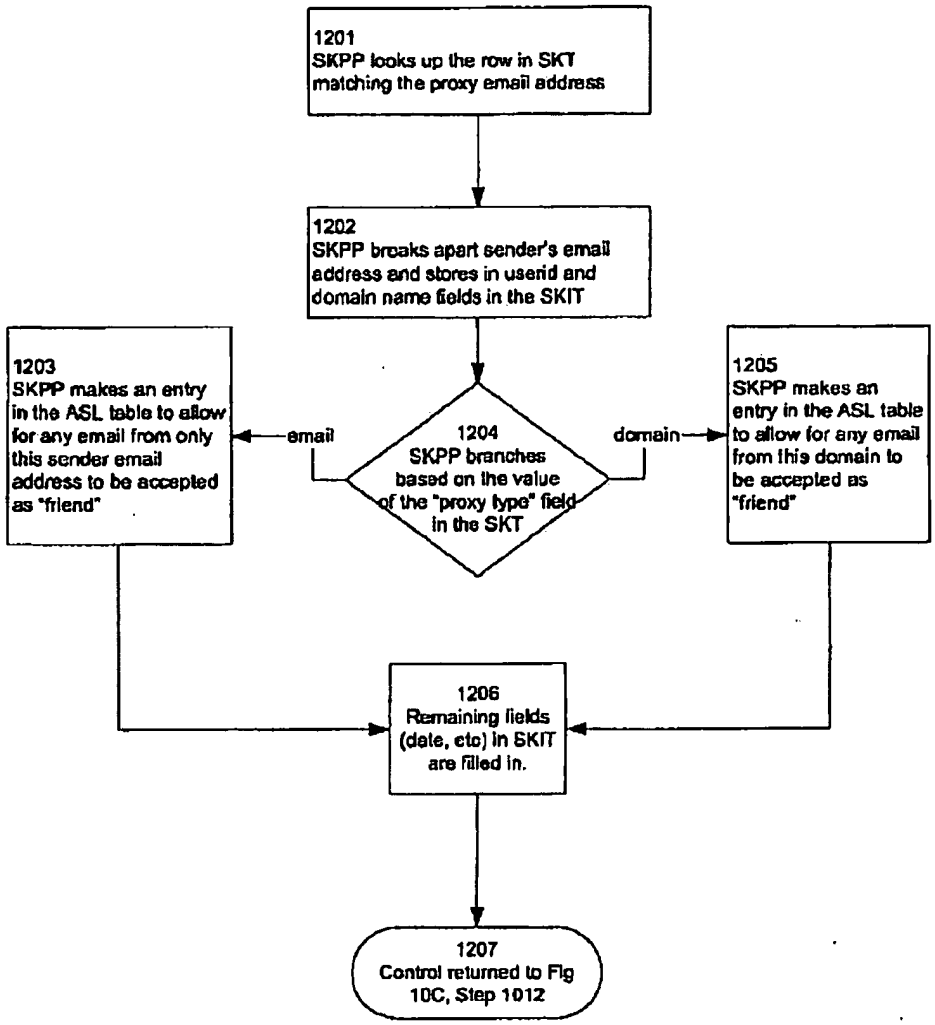
	a	b	c	d	e	f	g
1	justforacme@spamkapu.com	joe@spamkapu.com	12/20/2001	relative	720	1	domain
2	financialtimes@spamkapu.com	joe@spamkapu.com	12/20/2001	relative	720	2	domain
3	9874351@spamkapu.com	joe@spamkapu.com	12/27/2001	absolute	720	5	email
4	456789@spamkapu.com	joe@spamkapu.com	12/20/2001	absolute	720	1	email
5	localpapers@spamkapu.com	joe@spamkapu.com	1/3/2002	relative	0	2	domain
6	amazonorder@spamkapu.com	joe@spamkapu.com	2/15/2002	absolute	0	3	email

Figure 11B:

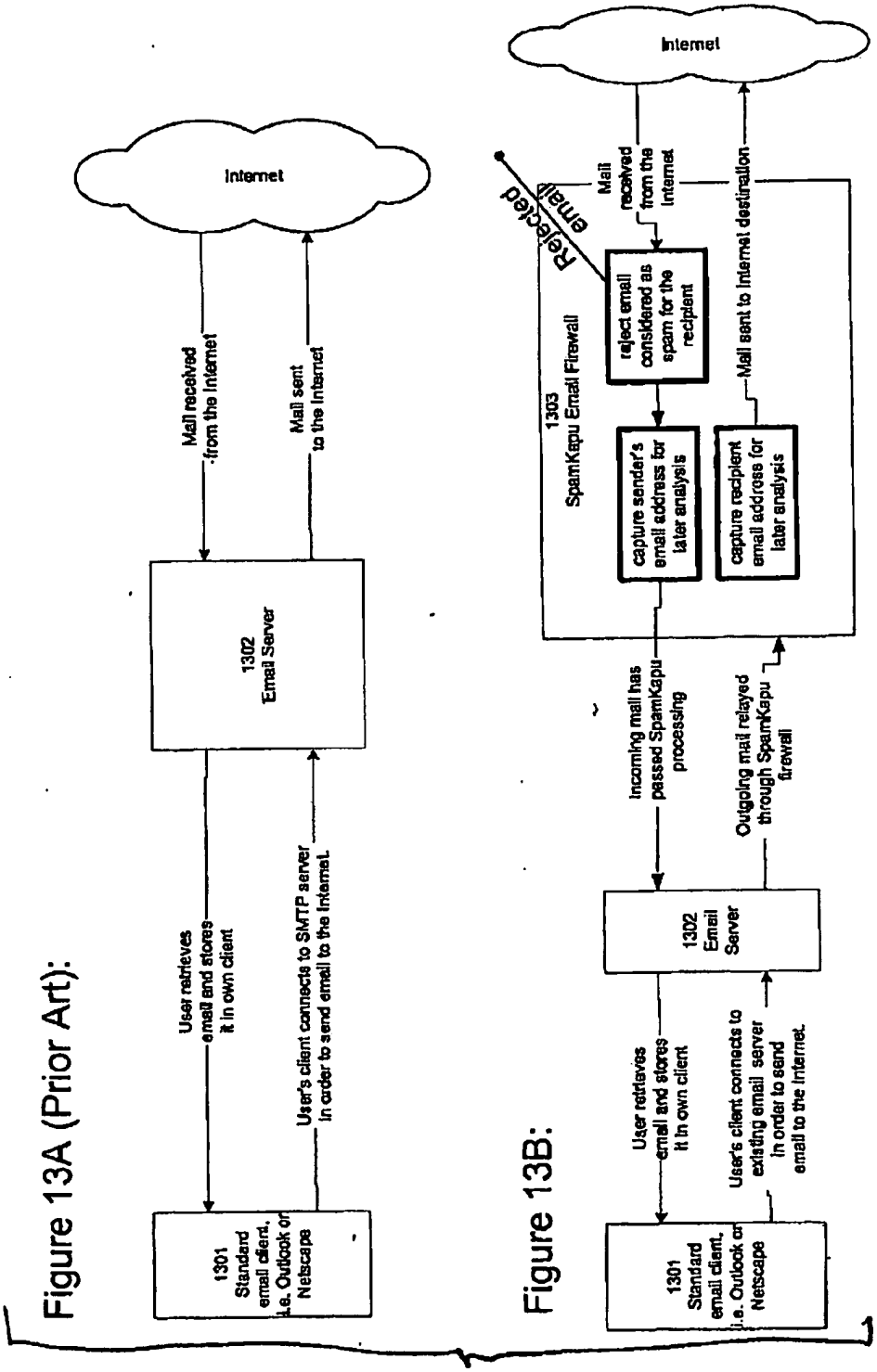
	a	b	c	d
1	justforacme@spamkapu.com	3/31/2003	tom	acme.com
2	financialtimes@spamkapu.com	11/6/2002	confirm	subscribers.com
3	financialtimes@spamkapu.com	6/5/2002	newsletters	subscribers.com
4	9874351@spamkapu.com	2/15/2002	harry	lava.net
5	9874351@spamkapu.com	2/3/2003	rich	isp.com
6	9874351@spamkapu.com	6/28/2002	sally	yahoo.com
7	9874351@spamkapu.com	10/6/2002	karen	hotmail.com
8	9874351@spamkapu.com	8/14/2002	mike	listit.com
9	456789@spamkapu.com	7/14/2002	sales	yourshoes.com

All new to CIP

Figure 12:



All new to CIP



Add new fo CIP

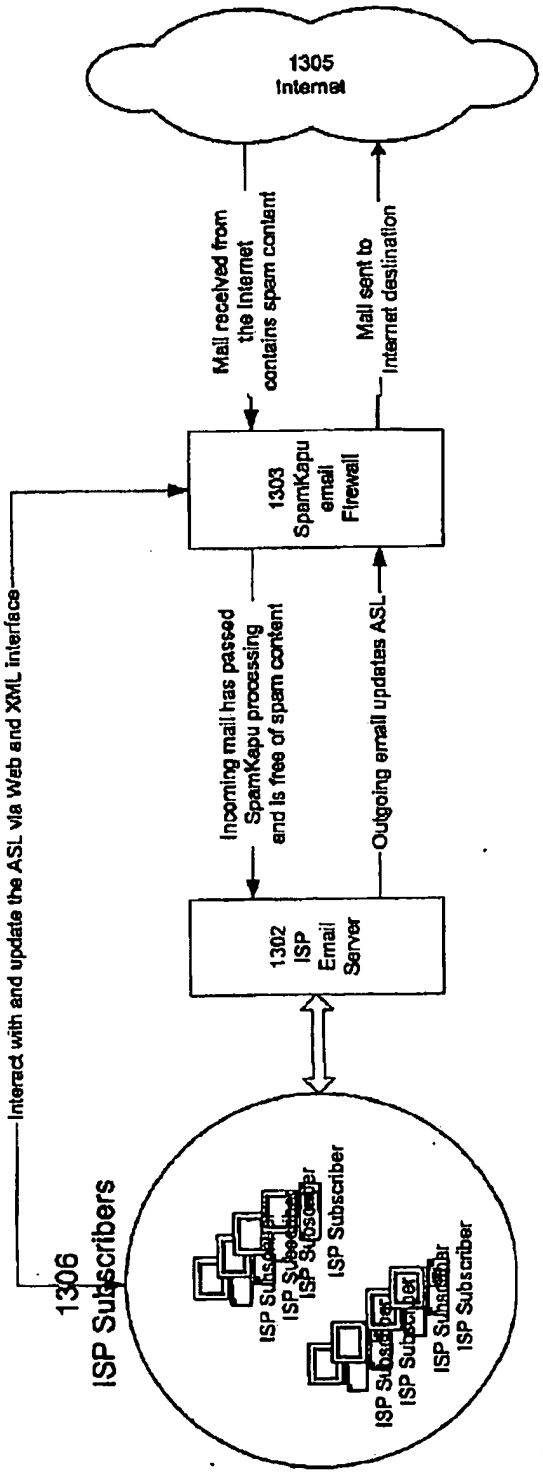


Figure 13C: All new to CIP

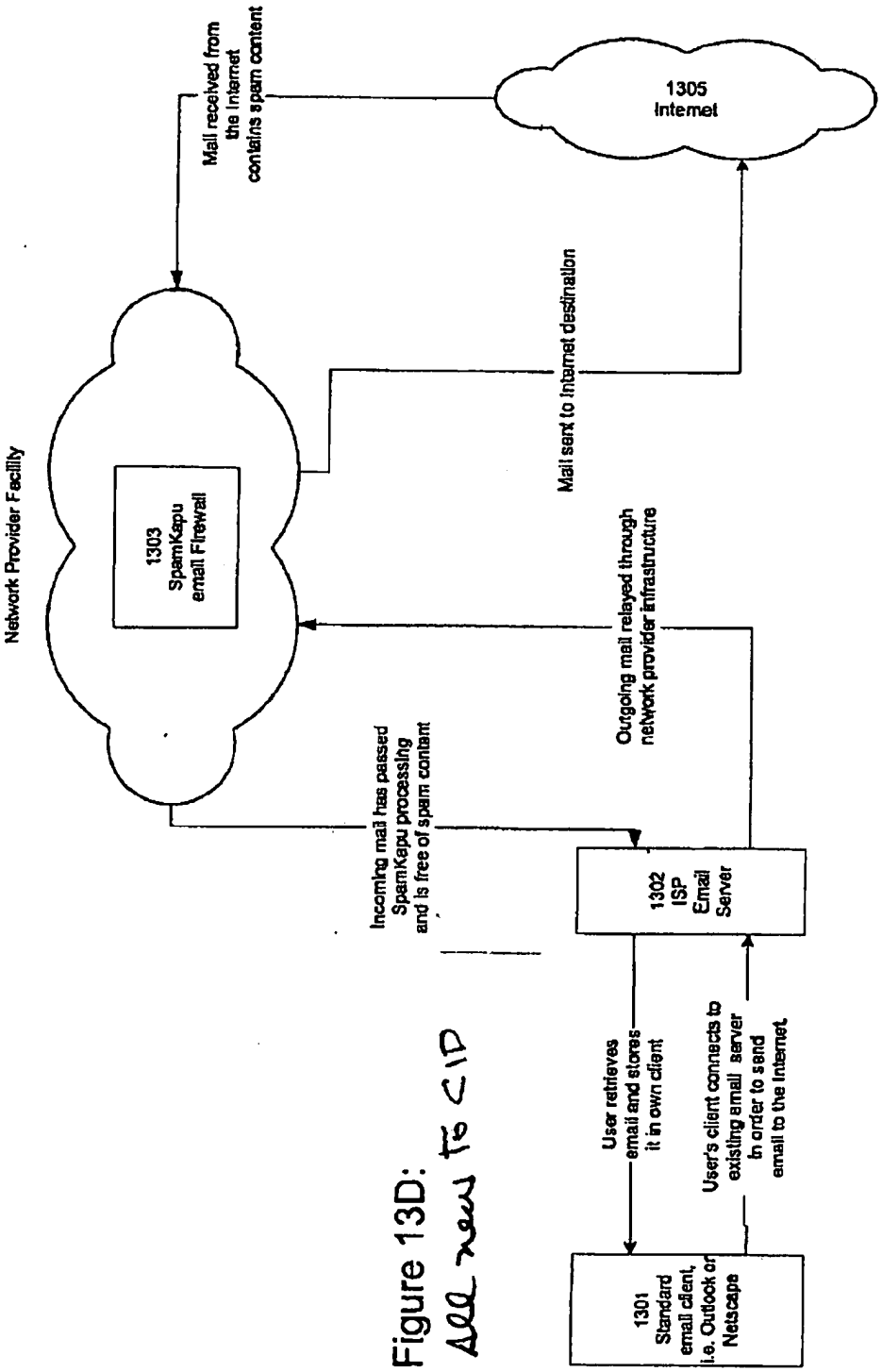


Figure 13D:
Add new FS <ID

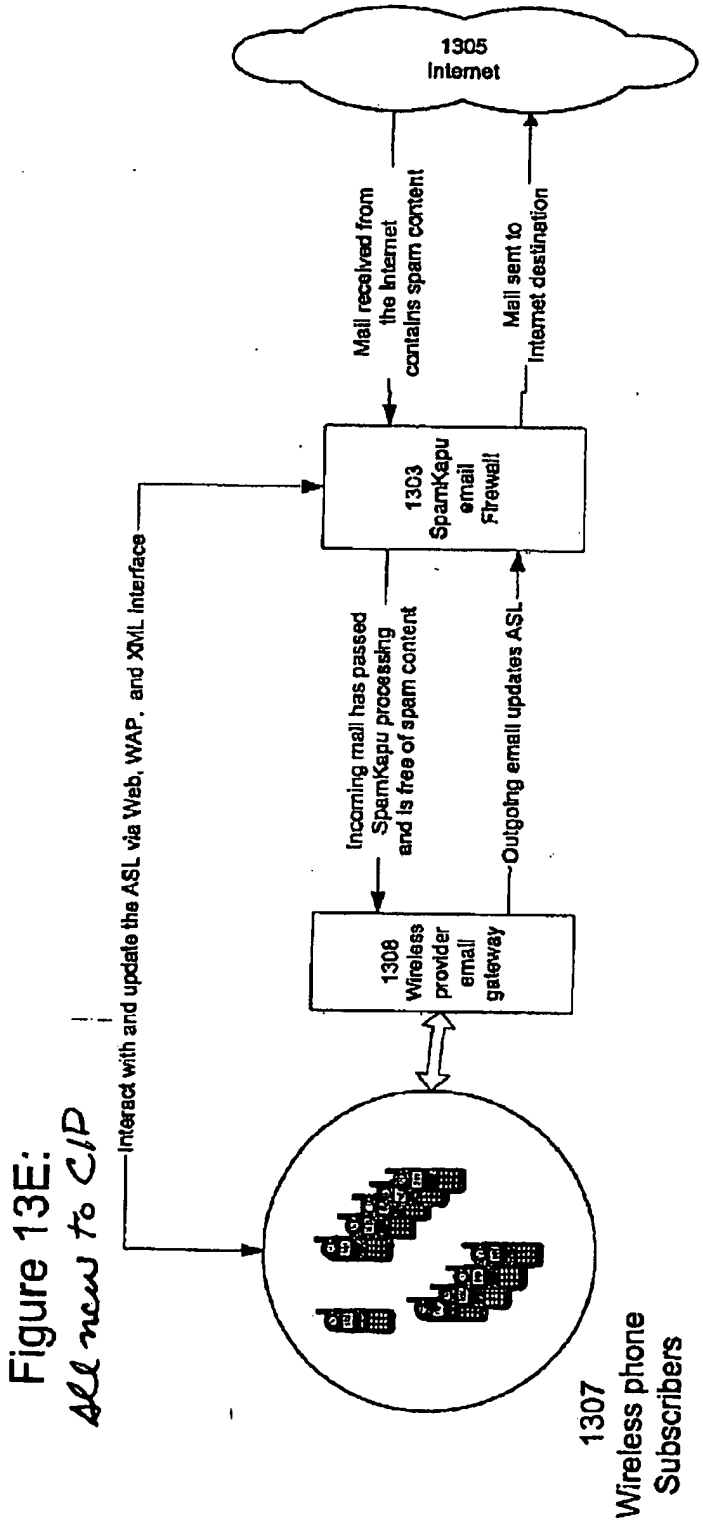


Figure 13E:
All new to CIP

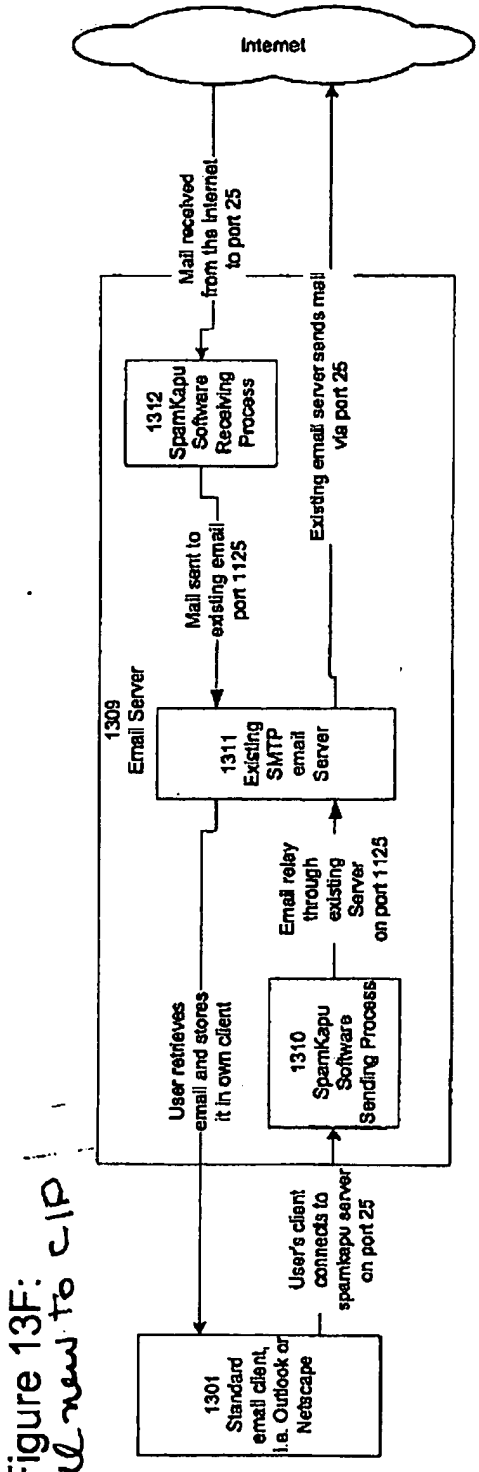


Figure 13F: See new to CIP

SYSTEM FOR ELIMINATING UNAUTHORIZED ELECTRONIC MAIL

[0001] This continuation-in-part U.S. patent application claims the priority of U.S. patent application Ser. No. 09/648,894, filed on Aug. 25, 2000, entitled "System for Eliminating Unauthorized Electronic Mail", which claimed the priority of U.S. Provisional Application No. 60/150,025, filed on Sep. 1, 1999, entitled "Unwanted Email Filtering System", and U.S. Provisional Application No. 60/180,937, filed on Feb. 8, 2000, entitled "Unwanted Email Filtering System", all by the same inventor.

FIELD OF THE INVENTION

[0002] This invention relates to a system for eliminating unwanted email, and particularly to one in which all email must be recognized as sent by an authorized sender in order to be accepted.

BACKGROUND OF THE INVENTION

[0003] Unwanted or unauthorized email is a significant bane for users on worldwide networks, such as the current public Internet. Once a person's email address becomes known in a network system, it can easily be replicated in computerized lists and passed on electronically to an unlimited number of parties who have not been authorized or invited to send email to the user. A user's electronic mailbox can become inundated with such unauthorized email. Unauthorized or unwanted email is referred to generically in the industry by the term "spam", although the term is not intended to be associated with or to disparage the popular canned meat product sold under the trademark "Spam" by Hormel Corp. The user may have an email address with a commercial information service provider (ISP) service which limits the amount of email that can be accepted and/or stored or which charges the user by the volume received. The user may also waste a significant amount of time opening and reviewing such unwanted email. Unauthorized email may also be sent by unscrupulous persons who may enclose a virus or noxious software agent in the email which can infect the user's computer system, or which can be used as an unauthorized point of entry into a local network system that handles the user's email.

[0004] Most, if not all, of the current software to control the receipt of spam is based upon the use of identifying lists of known spam sources or senders ("spammers"). Such conventional spam control software functions on the basis of receiving all email as authorized unless a sender is identified as being on the exclusion list and the email can be filtered out. This approach is only as good as the identifying list and cannot guarantee that the user will not receive spam. Spammer lists require frequent updating and must be distributed in a timely manner to all subscribers to the spam control software or service. Sophisticated spammers frequently change their source Internet address, and can defeat attempts to keep exclusion lists current. They can also route the unwanted email through the Internet servers of other parties so as to disguise the source of the emails through innocuous or popularly recognized names. A user's email address may also become known to large numbers of individuals in public chat rooms or on public bulletin boards. Unwanted email sent by individuals are not tracked on spammer lists, because the sending of email by individuals is technically not spamming.

SUMMARY OF THE INVENTION

[0005] Accordingly, it is a principal object of the present invention to provide a spam control system that cannot be defeated by spammers who frequently change their source addresses or disguise themselves by routing email through other servers, or by individuals who send email that are not invited or authorized by the user. It is a particular object of the invention that the system of the invention rejects all email as unauthorized unless the sender is recognized as being on the user's acceptance list.

[0006] In accordance with the present invention, a system for eliminating unauthorized email sent to a user on a network comprises:

[0007] (a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

[0008] (b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

[0009] (c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any incoming email addressed to the email address of the user.

Revised from 594 parent app.

[0010] In a preferred embodiment, the system's ASL module includes an ASL rules database for storing ASL rules lists of authorized sender addresses and associated processing rules for respective subscribers of the system, a spam processor module for processing the ASL rule list for matches, and an ASL manager for creating, maintaining, and updating the ASL rule lists. A redirector module rejects email based on the outcome of the spam processor module processing the sender's address against the ASL rule list. Email rejected by the redirector module is redirected to a web-based messaging (WBM) website and a message is sent notifying the sender to visit the WBM site and confirm that the sender is a legitimate sender of email to the intended recipient. If the sender logs on to confirm their status, the WBM component on the site executes an interaction procedure which can only be performed by a human, in order to ensure that the confirmation procedure is not performed by a mechanical program. The ASL manager maintains the ASL rule lists based upon sender address data collected from various sources and analyses of various email usage factors, including sent email, received email, contact lists maintained by the user, user preference inputs, third party programs, etc.

[0011] The invention also encompasses associated methods of performing the above functions, as well as related software components which enable these functions to be performed.

[0012] Other objects, features, and advantages of the present invention will be described in further detail below, with reference to the following drawings:

BRIEF DESCRIPTION OF DRAWINGS

[0013] FIG. 1A is a block diagram illustrating a standard Internet email system using the conventional method for

filtering email from spammers (Prior Art), as compared to FIG. 1B which shows a conceptual overview of a system in accordance with the present invention.

[0014] FIG. 2 is a process flow diagram for a preferred embodiment of the anti-spam system of the present invention.

[0015] FIG. 3A is a block diagram illustrating a standard SMTP send email process (Prior Art), as compared to FIG. 3B which shows a modified send email process used in the present invention.

[0016] FIG. 4A is a block diagram illustrating a standard SMTP receive email process (Prior Art), as compared to FIG. 4B which shows a modified receive email process used in the present invention.

[0017] FIG. 5 is a process flow diagram illustrating the operation of an anti-spam processing routine in the preferred embodiment of the invention.

[0018] FIG. 6 is a process flow diagram illustrating the detailed operation of a Web-Based Messenger (WBM) routine for handling email initially rejected by the anti-spam control.

[0019] FIG. 7A is a block diagram illustrating a standard SMTP send-receive email handling process (Prior Art), as compared to FIG. 7B which shows a modified Redirector process for handling received email.

[0020] FIGS. 8A to 8D are schematic diagrams illustrating the structure and operation of the ASL Manager in the preferred embodiment of the spam control system.

[0021] FIG. 9 illustrates a detailed implementation of examples of processing of email send/receive and user contact data into specific forms of actions taken by the ASL Manager.

[0022] FIGS. 10A to 10C are schematic diagrams illustrating the structure and operation of the invention's email proxy address subsystem processing in the preferred embodiment of the spam control system.

[0023] FIGS. 11A and 11B illustrate examples of the implementation of processing rules and results associated with the email proxy processing subsystem.

[0024] FIG. 12 illustrates a detailed implementation of how the email proxy processing subsystem converts or instantiates incoming email addresses that have not been previously received.

[0025] FIGS. 13A to 13F are schematic diagrams illustrating how the invention in its preferred embodiment would be installed/configured in existing email server architectures.

DETAILED DESCRIPTION OF INVENTION

[0026] In contrast to the known approaches of existing spam control methods of accepting all email unless listed on an exclusion list as unauthorized, the fundamental principle of the present invention is to reject all email unless the rules processing returns a favorable response. In this manner, it is possible to filter out email that comes from unrecognized spammers as well as individuals who send email that is uninvited by the user. Unlike the known email filtering systems, the present invention does not attempt to filter out the unwanted email after it has been accepted. Rather, it outright rejects the email at the earliest entry level by returning a server-level "no such user" error message to the device that is transmitting the sender's email. Thus, the invention operates on the premise that all email will be preprocessed according to pre-set rules before the validity of the recipient's (user) email address will even be accepted as correct. This provides an inherently powerful and 100% effective spam control solution in an environment where spammers can instantaneously change their source address or apparent identity and individuals in public areas can obtain email addresses of other users and send them unwanted email.

[0027] The following is a detailed description of one preferred embodiment of a system for implementing the invention concept. In this embodiment, the spam control system intelligently formulates the "authorized senders" rules list based upon user-defined actions previously stored in the email proxy preprocessor, an ongoing analysis of the user's email usage, such as to whom and with what frequency sent email is addressed to other users, and through the gathering of high-level user contact data, such as a user's known contacts and associates identified on other lists or files maintained by the user which indicate persons considered as authorized. The "authorized senders" rules list may also be updated and manipulated by the user at any time to add or remove authorized senders and/or associated processing rules. While this specific implementation is used, and certain components are provided and configured to be interoperable in the described ways, it is to be understood that the full scope of the invention is deemed to encompass many other suitable modifications and variations to the described guiding principles of the invention.

[0028] FIG. 1A is a block diagram of a standard email system for sending and receiving email on the Internet and is used to explain the conventional method for filtering out email from spammers. The system follows a standard industry protocol for handling email on the Internet, referred to as SMTP. Users typically subscribe with a chosen ISP for Internet access and related services, including email services. The users access the Internet through the ISP using a dialup or high-speed line connection and a standard browser. The browser includes or functions with a standard email client 101, such as the Outlook TM email client distributed by Microsoft Corp., headquartered in Bellevue, Wash., or the Netscape™ email client used by AOL Online, Fairfield, Va. The ISP operates at a website address corresponding to its domain name which is addressable by users on the Internet. The ISP's service functions are performed for a large number of subscribers through one or more servers. Typically, an email server 102 is used to handle the email service functions. Email sent to the ISP from the Internet is received at SMTP Server 104, where various administrative functions are performed, such as checking whether the addressee is an authorized subscriber of the ISP, then the email is placed in a storage space reserved for that user, referred to as Inbox 103. When users connect to the ISP, they can retrieve their email and store it with their own email client (on their own computer). Users can send email by composing it locally at their email client, then uploading it to the SMTP Server 105 at the ISP, which then routes it to the recipient's email address on the Internet.

[0029] Conventional anti-spam control can be implemented with the SMTP Server and/or at the email client. Many ISPs implement an exclusion list of known spammers at the SMTP Server. In addition, they commonly allow a user to filter out unwanted email from certain senders known to the user. For example, the user's email client may have a filtering function that allows the user to input unwanted sender email addresses to the SMTP Server so that email received by the SMTP Server can be filtered out before being put into the user's Inbox. Further, independent software vendors sell sophisticated email handling programs that work with the user's email client. For example, some handling program have functions for categorizing received email into topical file folders, and email from unrecognized senders may be put into a "Miscellaneous" or "Unrecognized" file folder.

[0030] In FIG. 1B, a conceptual overview of a system in accordance with the present invention is shown. As before, the standard email client **101** is connected to an email server **102** for sending and receiving email to and from the Internet via SMTP Server **104** and Inbox **103**. However, in this modified operation, the present invention provides for an unauthorized-email rejection component **113** upstream of the existing email server which intercepts and rejects email before it is accepted by the email server. In the rejection component **113**, an Authorized Sender List (ASL) Manager captures recipient email addresses from email sent by the user, as shown at block **112**, and also captures sender email addresses from email sent to the user, as shown at block **107**. The ASL Manager analyzes the captured sender email addresses and recipient email addresses and employs certain pre-defined rules (described in further detail below) to add or remove email addresses from the "authorized senders" list, referred to as the ASL Rule List or Database. The ASL Rule List is used by the SPAMKAPU Server **113** to accept only email from senders that favorably pass the ASL processing and subsequently relays the email to the pre-existing standard SMTP email server **104** while rejecting all other email with a "not such user" error code, as indicated at block **109**.

[0031] Referring to FIG. 2, the process flow for the operational steps of the anti-spam system of the present invention will now be described. Certain terms used in the description are defined below:

[0032] SPAMKAPU: An example of the spam control system of the invention.

[0033] SUBSCRIBER: A person subscribing to an ISP email service that is using the spam control system of the invention.

[0034] FRIEND: An email-sending source that is authorized by the spam control system to send email to the SUBSCRIBER.

[0035] SPAMMER: An email-sending source that is not authorized to send email to the SUBSCRIBER, which is commonly understood to be an unknown or unauthorized party that is using a manual or computerized email list mailing program to send large volumes of emails repetitively through the Internet.

[0036] UNKNOWN: An email sending source that has not yet been identified as either a SPAMMER or a CONTACT.

[0037] Email sent from the Internet (**106**) is sent to the email address of the ISP for the SUBSCRIBER, referred to in block **201** as the SpamKapu Email Address (SKE). Received email must first pass through the skproxy preprocessor **202**. The skProxy preprocessor examines the "to:" email address against a table of proxy addresses and if there is a match appropriately processes the email before passing it to the Redirector **203**. The Redirector **203** sends a request for validation for the email from the Spam Processor **204** which maintains the Spam Processing Database (SPDB) **205**, including the Authorized Senders Rules List (ASL) **206**. The SPDB Database and ASL Rules List are the heart of SPAMKAPU, as they contain the processing rules and lists of persons authorized to send email to the respective SUBSCRIBERS of the system. The Spam Processor **204** sends a response, either that the sender's address on the email is not authorized on the ASL List, i.e., is a SPAMMER, or is authorized on the ASL rules list, i.e., is a FRIEND, or is not present at all on the ASL rules list, i.e. is a UNKNOWN. If the response is that it is a SPAMMER, the Redirector **203** rejects the email, as shown at block **207**, such as by sending a standard error message to the sending server that the user as addressed does not exist.

[0038] As a refinement to the system, a Web-Based Messenger (WBM) process at block **208** may be set up to provide a corrective procedure in the event that the rejected email is from someone not yet listed on the ASL list and therefore an UNKNOWN. The unauthorized email may actually be from a person who has not been previously processed in the anti-spam system but who has a legitimate reason to reach the SUBSCRIBER. The WBM process **208** is set up as part of the spam control system to which the rejected email is redirected. The WBM process then sends an email to the email sender, who is now treated as an UNKNOWN. For example, the email message may read:

[0039] "An email sent by you to SUBSCRIBER's address was redirected to this site as being sent from an unrecognized sender address which may be a source of spam email. If you would like to confirm yourself as a person with legitimate reason to reach the SUBSCRIBER, please visit the WBM site and confirm your status as a FRIEND."

[0040] The WBM may have a separate web site address for interactions with UNKNOWNS. When an UNKNOWN receives the error response email, if they are a legitimate FRIEND for the SUBSCRIBER, they may elect to go to the WBM site to confirm their status as a legitimate FRIEND. If done before the expiration date, the WBM process will add an entry into the ASL rules list so that the now validated FRIEND may re-send the previous email and send future emails without error at shown in block 209. If the SUSPECT does not respond, this fact is also sent to the ASL Manager for further analysis. The extra confirmation step effectively eliminates SPAMMERS since they use automated programs to send out batch email and typically will not take human response time to log on to the WBM site to confirm their legitimate status.

[0041] If the Spam Processor sends a validation response that the sender is a FRIEND, then the Redirector 203 passes the email to the designated existing SMTP server 211 which processes the email accordance with existing Internet standards (RFC821). The user can now collect their email their

Revised

standard Inbox 212 (using standard Internet protocols such as POP3 or IMAP4) through the user email client 101 on their computer. Their email is 100% spam-free, since all email from senders not recognized by the system as authorized has been rejected.

[0042] Users send email composed on and sent from the email client 101 via standard SMTP protocols to the ISP's email server. The ISP's SMTP server is responsible for providing users with email addresses within the system, and sending users' email to the recipients' email addresses on the Internet 103. In the SPAMKAPU invention system, an SMTP Send Manager 214 is provided to intervene in the usual send email process. The SMTP Send Manager 212 copies header information from all outgoing email and sends the data to the ASL Manager 213, then sends the email on to the ISP's existing SMTP server which in-turn sends the mail to its intended destination as shown in block 215. The ASL Manager 213 performs one of the key functions in the invention system. It analyzes the header data from sent email and data from other data sources 216 maintained by the ISP email server system, such as email logs and user-supplied lists. On the basis of its analysis routines (to be described in further detail below), the ASL Manager 211 checks, populates, and updates the SPDB Database and ASL Rule List with the email addresses and other data on senders authorized to send email to the SUBSCRIBERS. The SPAMKAPU system also includes User Maintenance Modules (UMM) 217 which allows the user to interact with and upload user information to SPAMKAPU for further customization of SPAMKAPU's email operations for the user.

[0043] Referring to FIGS. 3A and 3B, a standard SMTP send email process (Prior Art) is shown compared to a modified send email process used in the present invention. In the standard send email process, in FIG. 3A, email sent from the user's email client to the ISP's email server may be pre-processed, such as checking for correct syntax, alias expansion, etc., and to identify the list of recipient email addresses (could be 1 or more). The server email manager gets each recipient email address in turn and attempts to establish a connection to the destination SMTP server and verify if the recipient email address is proper. If negotiation is unsuccessful, an error message is returned to the sending SMTP server. If negotiation is successful, the sending server sends the message body to the destination server and performs a proper "close connection" operation. In the modified send email process of the invention, in FIG. 3B, the email sent from the client is pre-processed by the SPAMKAPU SMTP Send manager 214 which copies the all recipient email address(es) including but not limited to the "TO: CC: and BCC:" addresses and sends the data to the ASL Manager 213. The SPAMKAPU SMTP Send Manager then passes the email to the existing ISP email server for transmission to the actual destination(s). On the assumption that the SUBSCRIBER authorizes email to be received from any person the SUBSCRIBER has sent email to, the proper email addresses of persons to whom the SUBSCRIBER has sent email are added to the ASL list of persons authorized to send email to the SUBSCRIBER. The sent email data can be used in further analyses by the ASL Manager, e.g., to upgrade a person's authorized status from temporary to permanent if more than a threshold number of email is sent by the SUBSCRIBER to the same person.

[0044] Referring to FIGS. 4A and 4B, a standard SMTP receive email process (Prior Art) is shown compared to a modified receive email process used in the present invention. In the standard receive email process, in FIG. 4A, email is received by the SMTP server from sender sources on the Internet and the server stores the email in the user's Inbox. In the modified receive email process of the invention, in FIG. 4B, the received email is subjected to processing by the skproxy and redirector as shown in blocks 202 through 206 to determine the nature of the sender's address (FRIEND, SPAMMER, or UNKNOWN). Even though the sender is already on the ASL authorized persons list, the received email data can be used in further analyses by the ASL Manager, e.g., to upgrade a person's authorized status from temporary to permanent if email from that person is received on an ongoing basis and has not been changed by the user. The SMTP receive email process then sends the email to the existing SMTP server via standard (rfc821) relay protocols for normal processing.

[0045] In FIG. 5, a process flow diagram illustrates the operation of the Spam Processor 203. At block 501, a request from the calling routine, here Redirector 203, seeks validation whether a received email is from an authorized sender. The request identifies the parameters who the email is FROM and who it is sent TO. The Spam Processor 204 uses the TO address to lookup that user's ASL list 206 in the SPDB Database 205, as indicated at block 502. The lookup procedure follows a loop 503 of reading the next ASL record on the user's ASL list, checking for a match to the email FROM address (authorized person), reading the next record if there is no match of the current record, executing the match condition by issuing a TRUE value if found, otherwise returning for the next record, as indicated at block 504. At block 505, if a TRUE VALUE is issued, then at block 505 the action is taken of executing the processes as defined in the Spam Processing Database or SPDB for this particular FROM/TO combination. Examples of processes include setting the output value to either FRIEND, SPAMMER, or UNKNOWN but also may include the execution of 3rd party software that may determine a FROM source is blacklisted or even determining that the email contains viruses. At block 506, the returned value is sent as a message to the calling routine, i.e., the Redirector 203. If the returned value is UNKNOWN, a standard error message is included. As a default option, if no ASL list is found for the user, the system returns the value FRIEND, as indicated at block 507, in order to allow the email to be accepted as a temporary condition until an ASL list can be established for that user. The request processing routine can be implemented using industry standard PERL programming syntax and incorporating a PERL interpreter to execute the processing rules.

[0046] In FIG. 6, a process flow diagram illustrates the detailed operation of the Web-Based Messenger (WBM) routine for handling email rejected by the Redirector 202 (see FIG. 2). Preferably, the WBM process is implemented via interaction with a rejected sender at a separate Web site address. In Phase 1, corresponding to step 204 in FIG. 2, the WBM process is initialized at block 601 by the ASL rule returning a value for rejecting an email as sent from an UNKNOWN by the Redirector 203. At block 602, a unique ID number is assigned to the UNKNOWN sender's email address in the WBM database and a given expiration date is set, e.g., 48 hours. At block 603, a return email is sent back to the sender's email address in order to notify the

UNKNOWN to go to the WBM web page if they wish to follow through with contacting the SUBSCRIBER. The WBM then waits for the UNKNOWN to go to the WBM site to complete the process, referred to as Phase 2. At block **604**, the UNKNOWN accesses the WBM web site and agrees to the displayed terms and conditions of usage. At block **605**, the WBM process verifies that the time for response for the email corresponding to the ID number has not expired. The WBM then follows a test procedure to ensure that the responding UNKNOWN is not being implemented by a mechanical program. For example, at block **606**, a word or question stylized in non-standard font can be displayed as a graphic image, and at block **607** the SPAMMER is prompted to type the word or answer the question that appears in the graphic. A mechanical program would not be able to read a graphic image of a word in unrecognizable font or would not be able to answer the question. At block **608**, if the WBM process determines that a correct word or answer has been typed, the UNKNOWN status is upgraded to FRIEND on the user's ASL Rule list. At block **609**, the WBM process notifies the FRIEND that he/she may re-send their original email and/or other email to the SUBSCRIBER. At block **612**, if the SUBSCRIBER determines that the email is from someone whose email should be rejected without a WBM error reply option, the SUBSCRIBER may optionally downgrade the status permanently to SPAMMER through the UMM **214**.

[0047] Referring to **FIG. 7A**, a block diagram illustrates a standard SMTP send-receive email handling process (Prior Art), as compared to **FIG. 7B** which shows a modified Redirector process for handling received email. In the standard process, the Sender-SMTP **701** requests connection to the Receiver-SMTP **702**, which accepts the connection if available. The Sender SMTP then performs the task in its Send Email loop of sending the recipient's email address. At block **703**, the Receiver-SMTP confirms or denies whether the recipient exists or whether it has authority to process email for this user. If confirmed, the Sender-SMTP sends the message body and marks the end of the message. At block **704**, the Receiver-SMTP receives the message body and sends it to the email box of the recipient (or recipients if the message is sent to more than one recipient at that SMTP server address.

[0048] In **FIG. 7B**, the Sender-SMTP **701** and Receiver-SMTP **702** perform their usual establishing of a connection and check for valid recipient e-mail address. However, in this modified process implemented in conjunction with the Spam Processor **705**, the sender's email header information, including the FROM address is stored by the Spam Processor for later use, as indicated at block **706**. At block **707**, the sender's FROM address and the recipient's TO address are sent to the Spam Processor **705**, by a request for validation by the Redirector as described previously. At block **708**, after checking the recipient's ASL Rules list to determine the status of the sender, the Spam Processor can return a response of FRIEND or a response of SPAMMER or UNKNOWN with an accompanying error message. If the response is FRIEND, an output is sent to the Sender-SMTP confirming that the email can be received, and the email is sent to the Receiver-SMTP as usual. At block **709**, the Receiver-SMTP relays the email to the recipient's email server for standard inbox processing and, if desired, can include a message noting that the sender was identified on the ASL list as a FRIEND. If the response is SPAMMER, then an error message is returned to the Sender-SMTP that the recipient does not exist or the Recipient-SMTP is not authorized to accept the email. If the response is UNKNOWN, the Receiver-SMTP may send the email through the WBM process, as described previously (indicated at block **710**), if the response from the Spam Processor indicates that the status of the sender is an UNKNOWN sender (as opposed to having the confirmed status of SPAMMER).

[0049] In **FIG. 8A**, a schematic diagram illustrates the structure and operation of the ASL Manager, previously described as component **211** with respect to **FIG. 2**. The ASL Manager preferably is structured to have an ASL On-Demand Processor **801** and an ASL Scheduler Processor **802**, both of which utilize industry standard XML and Web service protocols to interact with an ASL Rules Processor **806**, which also exchanges data with the Spam Processor Database (SPDB) **205**. Email addresses sent to and received from the SMTP Send Manager **214** and SMTP Receive Manager **211** are processed by the ASL On-Demand Processor **801** which executes the appropriate rules in conjunction with the ASL Rules Processor **803**. Content from a variety of other sources, including compatible third party plug-ins, can also be processed to create, populate, and update the ASL lists stored in the SPDB **205**. For example, content may be received from a "Drag and Drop Manager" for conveniently handling user address inputs while working with the email client, user address inputs from Web sites while working with an associated browser, addresses added by the user to a desktop contact manager, such as the Microsoft Outlook TM Address Book, or other contact lists, and other address inputs generated by third party software that can operate with the user's client programs.

[0050] The ASL Scheduler Processor **802** is used to process tasks on a scheduled basis for various analysis and maintenance functions. This allows a very rich examination of the SUBSCRIBER's ASL list, mail log, and other data files, to continually refine the "authorized senders" list for accuracy and relevance. For example, the processor functions can include: an ASL Mail Log Analyzer for analyzing the ASL Mail Log database **804** of the SUBSCRIBER's received and sent emails; an Expiration Date Analyzer for setting and enforcing expiration dates for authorized senders to be re-authorized; a Low Volume Analyzer for downgrading or eliminating the authorization status of senders with whom the SUBSCRIBER communicates very infrequently; a High Volume Analyzer for upgrading or permanently marking the authorization status of senders with whom the SUBSCRIBER communicates very frequently; a Fuzzy Logic Analyzer for making qualitative decisions as to FRIEND or SPAMMER status based on a variety of factors; and other Third Party Analyzers for analyzing data generated by third party plug-ins and programs to refine the ASL list.

[0051] The ASL Rules Processor **806** contains the rules (in an ASL Manager Rules Database **803**) that determine how to add, update or modify the ASL Lists maintained in the SPDB Database **205**. The Rules Processor can have an architecture that readily accepts and interoperates with third party databases **805** and applications programs **807** in order to harness the collective power of developers in the network communications industry to continually improve and extend the SPAMKAPU system's feature set. The ultimate result of this

architecture is to enable the creation of a very richly detailed ASL database which goes beyond even the total elimination of spam email into other or future needs of users for the dynamic and intelligent handling of email.

[0052] FIGS. 8B, 8C, 8D, illustrate various configurations where the invention interfaces with 3rd party services and software. FIG. 8B illustrates a specific example of how the on-demand processor operates. In this case, 3rd party software installed on the subscriber's client computer 810 gathers all email addresses stored in an application such as Microsoft Outlook, then connects to the spamKapu server using an XML interface 811 and uploads the contacts into the ASL Rules database 812, marking them as "Friend".

[0053] FIG. 8C illustrates a specific example of how the ASL Scheduled Processor invokes an XML interface 811 to connect to a remote 3rd party service 815 to perform a detailed analysis of the ASL Rules database. Updates to the database are transmitted back by XML interface 815 to update the ASL Rules database 812.

[0054] FIG. 8D illustrates a specific example of how the ASL Rules database can utilize a 3rd party to analyze email in real-time. As email is received by the SpamKapu server 818 an ASL Rule is invoked to used a 3rd party 819 and uses an XML interface 811 to connect to a 3rd party real-time email analysis service 821 which may employ sophisticated pattern matching analysis, for example. The service 821 uses an XML interface 811 to return a result of SPAMMER, FRIEND, or UNKNOWN 823 which is then further processed by the SPAM PROCESSOR 204.

[0055] In FIG. 9, a detailed implementation is illustrated of examples of processing of email send/receive and user contact data into specific forms of actions taken by the ASL Manager. The basic process flow consists of: Step 901 of looping through each line of an ASL rules list (called a Table) comparing the FROM address captured from an incoming email for a match; Step 902 of determining whatever condition or status flag has been set for the matched entry, then executing the corresponding condition rule as maintained on the Condition Table, resulting in return of a Return Value; and Step 903, based on the Return Value, executing the corresponding action rule as maintained on the Action Table, and exiting with a Final Return Value from this action. To follow one example through this process flow, Step 901 finds a FROM match of the sender address john@home.com, Step 902 notes the expiration date condition "before Dec. 1, 2003" and executes the "before" condition on the Condition Table to return a value of "True" if today's date is less than the indicated expiration date, and Step 903 notes that the sender status action (if condition is True) is "friend" and executes the "friend" action on the Action Table to return a Final Return Value of FRIEND (no parameters needed) as the validation response of the Spam Processor.

[0056] The specific programming syntax or execution logic of the ASL Manager rules processing may be varied in any suitable manner depending on the developer of the Spam Processor application. The following examples of some options for ASL Manager actions illustrate a wide range of approaches that may be used:

[0057] MATCHING AN EMAIL ADDRESS OR ADDRESS PATTERN:

[0058] (a) Default: exact match

[0059] (b) A specific email address: john@company.com

[0060] (c) UNIX Standard wildcard matching:

[0061] *.microsoft.com=anything from "Microsoft.com"

[0062] microsoft*=anything with microsoft in it

[0063] *.mil=any email from the military

[0064] (d) Matching any known "blackhole list" by using a %BLACKHOLE% symbol.

[0065] USING A CONDITIONAL AND PARAMETERS TO EXECUTE IF THE MATCH IS TRUE

[0066] USING A SECONDARY ACTION AND PARAMETERS TO PERFORM IF THE CONDITIONAL IS TRUE.

[0067] USING THE LAST DATE THE SUBSCRIBER SENT EMAIL TO THIS ADDRESS

[0068] USING THE LAST DATE THIS ADDRESS SENT EMAIL TO THE SUBSCRIBER

[0069] USING DATE THE RECORD WAS CREATED

[0070] EXAMPLES OF CONDITIONALS THAT CAN BE USED

[0071] (a) Expiration dates: use a given address until Feb. 12, 2004

[0072] (b) Date ranges: use a given address from Apr. 1, 2004 to May 2, 2004

[0073] (c) Specific recurring times: first week of every month but no other time, e.g., newsletter@magazine.com acceptable during 1st week of each month.

[0074] (d) A link to external software designed to allow for additional user-defined criteria; this allows for third party applications

[0075] EXAMPLES OF MESSAGES THAT MAY BE INVOKED BY A GIVEN SECONDARY ACTION

[0076] (a) Standard "error"

[0077] (b) Custom with variable substitution in the message body, e.g.:

[0078] %username% is substituted with the sender's email address

[0079] %subid% is the ID code of the subscriber

[0080] %date% is today's date

[0081] (c) "hello %username% you have been identified as spam, go to http://www.spamkapu-.com/subscriber=%subid% and if you're really human we'll let you in.

[0082] (d) Custom text: "All email addresses from America Online are unconditionally rejected"

[0083] (e) Send a given message in the error response.

[0084] (f) Send a given message as an email.

[0085] (g) Open a file and email its contents

[0086] (h) Open a file and send its contents as an error reponse.

[0087] (i) Set the sender's status to SPAMMER or FRIEND

[0088] (j) Create a unique ID that will expire after a short time period (24-48 hrs). This id can be used by the SUSPECT to access the WBM and become a CONTACT.

[0089] (k) Give SMTP default error message

[0090] (l) Link and execute external software designed to allow for additional user-defined actions; this allows for third party applications.

New

[0091] In FIGS. 10A and 10B, a schematic diagram illustrates the structure and operation of the SkProxy email preprocessor. The preprocessor simplifies the method by which users can manage their ASL rules list in situations where the user may wish to receive email from source whose "FROM:" email address is not known. Examples of such include subscription to newsletters where one supplies their email address to a newsletter service without knowing the correct "FROM" address until the newsletter is actually received, and online ordering procedures that request a user email address for the purposes of sending a confirmation email but do not disclose their "FROM" email address. The preprocessor is an additional software module that is designed to reside on the same server as the other SpamKapu software. All incoming email is first processed by the SkProxy processor and then passed on to other SpamKapu software modules as warranted.

[0092] FIG. 10A illustrates the first phase of the general SkProxy process whereby in step 1001 a user uses a standard Web browser to connect to the SpamKapu server, then requests and is given a list of "Proxy" email address which do not directly disclose or identify the user's actual real email address step 1002. As an alternative, the user may also first create their own proxy email address, then enter that address into the SKT (FIG. 11A) via a Web interface. This liberates the end-user from having to obtain email addresses beforehand. These Proxy email addresses along with their characteristics are stored in a table shown in FIG. 11A. hereinafter referred to as the "SKT".

[0093] In the second phase shown in FIG. 10B the end-user gives out these Proxy email addresses step 1003 as needed to subscribe to newsletters or other situations where the "FROM:" email address is not known. When the entity that has received this proxy email address wants to send an email to the end-user, they can only send it to the proxy email address step 1004 because that is the only address they have on file. The SkProxy process is the first to receive the email in Step 1005.

[0094] Control is then passed to FIG. 10C, step 1007, decide if the proxy address has been instantiated or not by examining the SkProxy Instantiation Table, described in FIG. 11B and hereinafter referred to as the SKIT, and use the proxy and sender addresses to find a match in the SKIT. If an entry does not exist (meaning that it has not been instantiated), then in step 1013, SkProxy counts the number of number of senders using this proxy address by examining the SKIT and determine if the number of instantiations has reached the maximum allowed as defined by the user in the SKT. If the maximum has not been reached, it instantiates the proxy address by creating an entry in the SKIT, as detailed in FIG. 12 and step 1012. In Step 1011, it references the entries in the SKT table along with the senders FROM address to create appropriate entries in the ASL Rules list as detailed in FIG. 12. In Step 1008, it rewrites and updates the "TO:" address to be the actual address of the end-user and in Step 1009 adds the proxy email address in the end-user name area so that the end-user can see what proxy email address was used by the sender. Step 1010 passes control to the Redirector which may now properly evaluate the From/To: combination and perform the correct processing. The email will now be accepted by the SpamKapu server because the instantiation process has created created appropriate ASL rules that will allow this email to pass through without requiring WBM validation.

[0095] If the maximum instantiations has been reached as determined by step 1013, SkProxy does not instantiate any further proxy entries and passes control to step 1010, returning to the Redirector. The net effect of disallowing further instantiations will be that no ASL rules will be entered and since it is highly likely that no ASL rules already exist with this sender's email address, the sender's email will most likely be rejected by the Redirector.

New

[0096] FIG. 11A illustrates a sample representation of the SkProxy Table ("SKT") used to hold SkProxy-related information. This table's function is to record all the proxy addresses available along with various characteristics that direct what kind of ASL entries the proxy address will create upon instantiation. The following describes each field:

Label in FIG. 11A	Column Description
a	Proxyaddress: the email address that can be given out by the end-user. Senders sending email to this address will reach the end-user if the proper conditions are met.
b	true email: the real email address of the SpamKapu end-user
c	creation date: the date this proxy address was created
d	expiration type: Relative or Absolute. A relative expiration will expire this proxy based on the number of dates elapsed from the date of a specific proxy address instantiation. An absolute expiration will expire this proxy based on the number of dates elapsed from the date of initial proxy address creation.
e	expiration days: the amount of days to allow to transpire. Used in conjunction with field d above to decide if a proxy address has expired.
f	max senders: the maximum amount of different senders that may instantiate use this proxy address.
g	proxy type: whether to create an ASL entry to allow the entire domain of the sender, or onl the specific email address of the sender

[0097] The following explains the sample data presented in FIG. 11A and describes their application in the SkProxy technology:

entire page new

Row	Example and description of use
1	The end-user will give it out to a person at a meeting with the intent of allowing anyone from that company to send email to the subscriber. When the first email comes from that person, the skProxy process will instantiate this entry, and make an entry in the ASL table to allow for any emails from this domain to pass through as "friend". The maximum senders allowable is set to 1, so the sender's domain will be the first and the only allowable domain (company) to use this proxy
2	End-user has given this proxy address to subscribe to a newsletter and has defined the proxy to allow 2 different senders to use the proxy email address.
3	End-user is creating a private mail network to execute a small project. Because the expiration was absolute, this proxy will expire 720 days after Dec. 27, 2002. Up to 5 senders may use the same proxy address.
4	The end-user will be providing his/her email address at a corporate Web site. When the first email comes from that site, the skProxy process will instantiate this entry, and make an entry in the ASL table to allow for any emails from that specific sender only to pass through as a "friend". Because max senders is set to 1, the first sender to instantiate the address will also be the only sender that can use that proxy address.
5	End-user has given this proxy address to subscribe to a newsletter and has defined the proxy to allow 2 different domains to use the proxy email address. The expiration is set to 0 which means this proxy address will never expire.
6	End user intends to place an online order, does not want the proxy to expire, and will allow up to 3 different senders to use the same proxy address

[0098] FIG. 11B illustrates a sample representation of the SkProxy Instantiation Table ("SKIT") used to hold specific proxy instantiation information. This table's function is to record each instantiation of the proxy address, especially the sender's email address. The following describes each field:

Label in FIG. 11B	Column Description
a	Proxyaddress: same as FIG. 11A, column a
b	instantiation date: the date this proxy address was instantiated for this sender
c	sender id: the information to the left of the "@" sign of the sender's Internet email addresses
d	sender domain: the information right of the "@" sign of the sender's Internet email addresses

[0099] The following explains the sample data presented in FIG. 1B and describes the details of various instantiations corresponding to the sample data provided in FIG. 11A:

Row	Corresponding row in FIG. 11A	Description
1	1	The sender tom@acme.com sent an email to the proxy address justforacme@spamkapu.com. This row was instantiated as a result and subsequently Tom's email went through to the end-user without requiring the WBM validation. Because the column f in the corresponding row in FIG. 11A for this proxy address shows a limit of 1 sender only Tom can send email to justforacme@spamkapu.

-continued

Row	Corresponding row in FIG. 11A	Description
		com. Anyone else sending an email to justforacme@spamkapu.com will receive an industry-standard "no such user" error message.
2	2	The proxy address "financialtimes@spamkapu.com" was given out by the end-user to subscribe to a newsletter. This newsletter service sends out a confirming email before starting the subscription. This row's instantiation is the result of receiving a confirming email from the newsletter service with a "FROM" address of "confirm@subscribers.com", which passed through to the end-user without requiring WBM validation.
3	2	This row's instantiation is the result of receiving the actual newsletter. A second email from the newsletter service with a "FROM" address of "newsletters@subscribers.com", which passed through to the end-user without requiring WBM validation. Subsequent newsletters have the same "FROM" address and therefore do not create additional instantiation records and also pass through without WBM validation. Because the max senders field in column f of the corresponding row in FIG. 11A is set to 2 and because there are 2 entries in this table, any other email sent to the proxy address "financialtimes@spamkapu.com", will not be instantiated in the SKIT table, no ASL rules will be created, and as a result the SpamKapu server will return an industry standard "no such user" error message.
4	3	Rows 4, 5, and 6 illustrate 3 different senders that are using the same proxy address. Emails from these senders will pass through without a WBM process. Because the corresponding entry in the SKT table shown in FIG 11A, row 3 column f, indicates that there are 5 max senders allowed, and there are only 3 different senders in this table, 2 more different senders may send an email to 9874351@spamkapu.com without requiring WBM validation. After the 2 additional instantiations have occurred, however, any other senders using the 9874351@spamkapu.com email address will receive an industry-standard "no such user" error message because no additional entries in the ASL table will be made.
5	3	See description for FIG. 11B Row 4 above.
6	3	See description for FIG. 11B Row 4 above.
7	4	The Web site that received a communication from the end-user has replied by sending an email FROM sales@yourshoes.com TO: 456789@spamkapu.com. This row is the resulting instantiation. Because column f in the corresponding row in FIG. 11A shows a max sender of 1, any other sender that attempts to send an email to 456789@spamkapu.com will receive an industry-standard "no such user" error message.
none	5	Note that there is an entry in FIG. 11A but there is no corresponding entry in FIG. 11 B. This is because no sender has sent an email to the proxy address in column a of the corresponding row in FIG. 11A.
none	6	Note that there is an entry in FIG. 11A but there is no corresponding entry in FIG. 11B. This is because no sender has sent an email to the proxy address in column a of the corresponding row in FIG. 11A.

[0100] FIG. 12 provides a detailed description of the instantiation process. "SKPP" is defined to mean "SkProxy

Entire page new

PreProcessor". In Step **1201** the SKPP searches for and matches the proxy email address (in the "TO:" field) in the SKT shown in **FIG. 11A**. Step **1202** extracts the sender's email address in to two parts, the "userid", represented by the information to the left of the "@" symbol in an Internet standard email address, and the domain, represented by the information to the right of the "@" symbol in an Internet standard email address. Step **1204** determines if the proxy is for an entire domain or for only a specific user email address. Step **1205** adds an entry in the ASL table to allow this and any further emails from this entire domain to be identified as "FRIEND" without further validation. An entire domain proxy would be useful if a user wanted the proxy email to be used by anyone within that domain; examples might include online orders or newsletters where it is acceptable to receive email from any address within that domain such as order-confirmation@amazon, order-status@amazon.com, order-support@amazon.com. Step **1203** adds an entry in the ASL table to allow this and any further emails from this specific email address only to be identified as "FRIEND" without further validation. A specific user address proxy may be useful if one wishes gives out an email address to a specific individual without forcing that individual to go through a validation process. In Step **1206**, an entry is made in the SKIT table shown in **FIG. 11B** and all the fields are filled. Step **1207** returns control to Step **1012** in **FIG. 10C** and the proxy processing process continues.

[0101] Once a proxy email address has been instantiated, it can only be used for the specific FROM domain or email address that it was instantiated for. For example, if an end-user submitted their domain-wide proxy email address for the purpose of an online order, and that email address was subsequently instantiated, the proxy email address cannot be successfully used by a sender that does not use the same domain name as the online order vendor.

[0102] By defining the maximum amount of senders that may use a given proxy address, the end-user can effectively create "private email networks" whereby the proxy email address will work for collection of individuals or organizations but does not work for others.

[0103] Existing standard email configurations as shown in **FIG. 13A** use an email server **1302** to receive email, store email in an inbox to be retrieved by client computer, and to transmit email sent by the client computer **1301**. The improvement in this invention as shown in **FIG. 13B** allows a SpamKapu server to be easily installed in an existing email network as a hardware network appliance device with minimal reconfiguration of the existing network and/or additional maintenance labor from staff.

[0104] **FIG. 13B** illustrates how the invention would be installed so that any incoming email would first pass through the processing of the SpamKapu server. Only validated FRIEND email would pass through to the (previously) existing email server **1302**. All existing configuration and processing on that existing email server would continue unchanged. Outgoing email would go from the client computer **1301** to the existing email server **1302** which in-turn will use industry-standard relay specifications to pass its email to the SpamKapu server **1303** which would copy all recipient email addresses to its ASL Rules List as "FRIENDS", then send the email to its intended destination server and recipient. This configuration provides a simple and transparent method to both block all incoming spam and easily copy the outgoing emailing addresses into the ASL Rules List.

[0105] **FIG. 13C** illustrates how the invention can be installed at an ISP facility to provide spam protection to the subscriber base. The spamKapu email firewall **1303** is installed to process mail addressed to subscriber accounts **1306**. Subscribers interact with the SpamKapu email firewall **1303** via the provided Web interfaces and XML interfaces (see **FIG. 8**). In this configuration, spam protection can be provided to an ISP subscriber base.

[0106] **FIG. 13D** illustrates how the invention can be installed at a network connectivity provider's facility to effectively offload spam-related email bandwidth for ISPs or corporate installations. All incoming email from the Internet **1305** is sent to the network provider facility **1304** that typically has significant bandwidth capability. All spam-related email is rejected by the SpamKapu email firewall **1303** as described previously. The remaining spam-free email **1307** then passed to the ISP or corporate email server **1302**. The end results is the effective reduction of bandwidth used to simply transmit spam.

[0107] **FIG. 13E** illustrates how the invention can be installed at a wireless telephone service provider facility to provide spam protection for wireless subscribers. Most carriers provide an Internet email gateway **1308** whereby wireless subscribers can send and receive Internet email. The spamKapu email firewall **1303** is installed to process mail addressed to subscriber accounts **1307**. Subscribers interact with the SpamKapu email firewall **1303** via the provided Web interfaces and XML interfaces (see **FIG. 8**). To provide added functionality for wireless phone subscribers, an additional communication layer that follows the WAP (wireless access protocol) is illustrated to allow subscribers to interact with the SpamKapu firewall using only their wireless device. In this configuration, spam protection can be provided an wireless subscriber base.

[0108] The firewall configuration is not intended to replace any existing firewall devices operating on the network. For network configuration purposes, the SpamKapu email firewall replaces the existing email server that processes external incoming email and transmits email addressed to external servers. The ideal configuration of the SpamKapu firewall is to be considered a hardware component alongside the existing email servers and in conjunction with any other existing firewall devices.

[0109] The optimal commercialization of the SpamKapu server will be as a network appliance. This can be packaged as a complete hardware and software solution or the software can be installed on dedicated hardware by knowledgeable technicians. The key envisioned commercial applications include A) Commercial ISPs that use the SpamKapu technology to provide spam elimination services to their clients. B) Network providers that offer both spam elimination and reduced bandwidth usage.

[0110] **FIG. 13F** illustrates how SpamKapu can be installed on an existing email server **1311** in a pure software configuration. The established Internet email standard protocol uses TCP/IP port 25 to send and receive email. By reconfiguring the existing email server **1311** to look for incoming email on another port, for example **1125**, and

transmit email on port 25 and configuring SpamKapu software 1312 to look for incoming mail on port 25 and send its mail through port 1125, the SpamKapu server is properly inserted into the email transmit and receive process. In the case of incoming email, it is first routed through SpamKapu's receive process in step 1312 via port 25 and then is sent to the existing email server via port 1125 where the user client 1301 can retrieve the email. In the case of outgoing mail, the client connects to SpamKapu's transmit process via port 25 which performs processing as detailed in FIG. 3*b*, and is then sent via port 1125 to the existing email server 1311 which in-turn sends the email to its destination over the Internet via port 25. The configuration illustrated in FIG. 13F allows the SpamKapu invention to be installed on existing server hardware thereby lowering the overall cost and maintenance involved with additional hardware.

[0111] In summary, the present invention provides a spam email rejection method which analyzes the sender address of incoming email and determines whether it is to be rejected before being accepted by an email-receiving server by returning a standard "no such user" error code or redirecting it elsewhere. This provides an advantage over existing anti-spam filtering systems which accept all email and attempts to filter out only those that have sender addresses recognized as those of known spammers. The invention employs an ASL module to capture authorized sender email addresses from the user's outgoing email or other sources in order to update the "authorized senders" (ASL) lists. The WBM procedure allows senders of rejected email to go to a separate website and register as valid senders after passing an interaction test that confirms that it is not being done by a mechanical program. The SkProxy procedure allows subscribers to use temporary proxy addresses for receiving email expected from unknown sources and instantiates senders as authorized upon receiving the expected email to the proxy addresses. The unauthorized-email rejection component of the system can be readily configured as a hardware or software appliance used in tandem with a conventional email server, email gateway, or firewall to an intranet, or as a software extension to an existing firewall system.

[0112] It is understood that many other modifications and variations may be devised given the above description of the guiding principles of the invention. It is intended that all such modifications and variations be considered as within the spirit and scope of this invention, as defined in the following claims.

I claim:

1. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user.

2. A system for eliminating unauthorized email sent to a user on a network according to claim 1, wherein the unauthorized-email rejection component is positioned in the flow of incoming email upstream from the email-receiving server such that unauthorized email is intercepted and prevented from reaching the email-receiving server.

3. A system for eliminating unauthorized email sent to a user on a network according to claim 1, wherein the unauthorized-email rejection component is configured as a hardware appliance that is positioned in the flow of incoming email physically upstream from the email-receiving server.

4. A system for eliminating unauthorized email sent to a user on a network according to claim 1, wherein the unauthorized-email rejection component is configured as a software component operable in the flow of incoming email logically upstream of the email-receiving server.

5. A system for eliminating unauthorized email sent to a user on a network according to claim 1, further comprising an email proxy pre-processing module that allows users to designate a destination proxy email address for use by a sender in instances where the email address of an authorized sender is not yet known, examines incoming email and, upon recognizing the destination proxy email address of the user, accepts the email and sends it to the user.

6. A system for eliminating unauthorized email sent to a user on a network according to claim 1, further comprising a WBM component for sending a message to a rejected sender inviting the sender to be validated as an authorized sender.

7. A system for eliminating unauthorized email sent to a user on a network according to claim 6, wherein the WBM component specifies a predetermined time period for the sender of rejected email to be validated as an authorized sender.

8. A system for eliminating unauthorized email sent to a user on a network according to claim 6, wherein the WBM component is installed on a separate website that can be accessed by a sender of rejected email to be validated as an authorized sender.

9. A system for eliminating unauthorized email sent to a user on a network according to claim 6, wherein the WBM component requires a sender of rejected email to pass an interaction procedure to show that the sender is not a mechanical program attempting to automatically validate the sender.

10. A system according to claim 9, wherein the interaction procedure includes a display of a graphic image of a word or object, and an input for the sender to enter in a text word in response to the graphic image, whereby the system can confirm that the interaction procedure is not being performed by a mechanical program.

11. A system according to claim 6, wherein once the sender is confirmed as an authorized sender of email to the intended recipient user, the WBM component sends an appropriate update to the ASL module that allows subsequent emails from the sender to pass through normally.

12. A system according to claim 1, wherein the ASL module includes an ASL database for storing ASL lists of processing rules and authorized sender addresses for respective users of the system, and a spam processor module for processing the ASL rules lists to determine if the sender is friend, spammer, or unknown.

13. A system according to claim 1, wherein, upon the ASL module determining that incoming email has a sender address that is not that of an authorized sender, said unauthorized-email rejection component rejects the incoming email with an industry standard "no such user" error message.

14. A system according to claim 1, wherein the ASL module includes an ASL list manager for analyzing email header information including FROM and TO addresses of email sent by users to dynamically update the ASL lists of authorized senders.

15. A system according to claim 1, wherein the ASL module includes a rules processor for processing authorized sender addresses for updating the ASL lists using data from an email address source selected from the group of email address sources consisting of: received email; sent email; user inputs to email service functions on the email client; inputs from user browsing of web sites; user desktop organizer and other contact lists; and third party address email management programs.

16. A system according to claim 1, wherein the ASL module includes a rules processor for processing analysis rules for updating the ASL lists using data from an analysis source selected from the group of analysis sources consisting of: user email log analysis; expiration date analysis; low/high email volume analysis; fuzzy logic analysis; and third party data analysis.

17. A system according to claim 1, wherein the ASL module maintains the ASL lists using a designation of sender-address status selected from the group of sender-address status designations consisting of: always authorized as a friend; authorized as a friend over a date range; authorized as a friend before an expiration date; authorizing all email for a given domain name; authorizing all email for a domain or any subdomain; always rejected as a spammer; rejected as a spammer matching a black list; status returned by executing a 3rd party email software; and rejected as a spammer sent with an error message.

18. A method for eliminating unauthorized email sent to a user on a network comprising the steps of:

(a) receiving incoming email addressed to the unique email address of the user,

(b) maintaining an authorized senders list (ASL list) of email addresses of external users authorized to send email to the user,

(c) processing the sender's email address on incoming email by comparing it to the ASL list, and

(d) rejecting the receipt of incoming email before the email can be accepted for delivery to the user if the results of processing the ASL list returns with a result of "unauthorized sender" by returning an industry standard "no such user" error code.

19. A method for eliminating unauthorized email sent to a user on a network comprising the steps of:

(a) receiving incoming email addressed to the unique email address of the user,

(b) maintaining an authorized senders list (ASL list) of email addresses of external users authorized to send email to the user,

(c) processing the sender's email address on incoming email by comparing it to the ASL list, and

(d) rejecting the receipt of incoming email before the email can be accepted for delivery to the user if the results of processing the ASL list returns with a result of "unauthorized sender", sending a message inviting the sender of the rejected email to confirm that the sender is an authorized sender of email to the intended recipient by passing an interaction procedure to show that the sender is not a mechanical program attempting to automatically validate the sender, and thereupon adding the validated sender's email address to the ASL list.

20. A method for eliminating unauthorized email sent to a user on a network comprising the steps of:

(a) receiving incoming email addressed to the unique email address of the user,

(b) maintaining an authorized senders list (ASL list) of email addresses of external users authorized to send email to the user,

(c) processing the sender's email address on incoming email by comparing it to the ASL list,

(d) rejecting the receipt of incoming email before the email can be accepted for delivery to the user if the results of processing the ASL list returns with a result of "unauthorized sender", and

(e) allowing a user to designate a destination proxy email address for use by a sender in instances where the email address of an authorized sender is not yet known, wherein if the destination proxy email address is recognized on incoming email, the incoming email is accepted and sent to the user.

21. A method for eliminating unauthorized email sent to a user on a network comprising the steps of:

(a) receiving incoming email addressed to the unique email address of the user,

(b) maintaining an authorized senders list (ASL list) of email addresses of external users authorized to send email to the user,

(c) processing the sender's email address on incoming email by comparing it to the ASL list, and

(d) rejecting the receipt of incoming email before the email can be accepted for delivery to the user if the results of processing the ASL list returns with a result of "unauthorized sender",

wherein the ASL module includes an ASL list manager for analyzing email header information including FROM and TO addresses of email sent by users to dynamically update the ASL list of authorized senders.

22. A method for eliminating unauthorized email sent to a user on a network comprising the steps of:

(a) receiving incoming email addressed to the unique email address of the user,

(b) maintaining an authorized senders list (ASL list) of email addresses of external users authorized to send email to the user,

(c) processing the sender's email address on incoming email by comparing it to the ASL list, and

(d) rejecting the receipt of incoming email if the results of processing the sender's email address with the ASL list returns with a result of "unauthorized sender", wherein the rejection of the incoming email is performed in a physical or logical operation before the email can be accepted for delivery to the user.

23. An unauthorized-email rejection component for use with an email-receiving server for receiving email sent to a user on a network comprising an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, wherein said unauthorized-email rejection component intercepts and rejects any incoming email addressed to the email address of the user if the processing results of the ASL list returns with an "unauthorized sender" result, wherein the unauthorized-email rejection component is a hardware or software appliance positioned for operation in the flow of incoming email physically or logically upstream from the email-receiving server to prevent any unauthorized email from reaching the email-receiving server.

24. An unauthorized-email rejection component for use with an email-receiving server for receiving email sent to a user on a network comprising an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, wherein said unauthorized-email rejection component intercepts and rejects any incoming email addressed to the email address of the user if the processing results of the ASL list returns with an "unauthorized sender" result, wherein the ASL module includes an ASL list manager for analyzing email header information including FROM and TO addresses of email sent by users to dynamically update the ASL lists of authorized senders.

25. An unauthorized-email rejection component for use with an email-receiving server for receiving email sent to a user on a network comprising an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, wherein said unauthorized-email rejection component intercepts and rejects any incoming email addressed to the email address of the user if the processing results of the ASL list returns with an "unauthorized sender" result, and further including a proxy address module for allowing a user to designate a destination proxy email address for use by a sender in instances where the email address of an authorized sender is not yet known, and if the destination proxy email address is used on incoming email, for accepting the incoming email and sending it to the user.

26. An unauthorized-email rejection component for eliminating unauthorized email according to claim 25, wherein upon receipt of incoming email using the destination proxy address, an entry for the email address of the sender thereof is dynamically created in the ASL module as an authorized sender.

27. An unauthorized-email rejection component for eliminating unauthorized email according to claim 25, wherein said unauthorized-email rejection processes the incoming email using the destination proxy address according to rules that permit email to be accepted from a specific user, domain, and/or subdomain represented in the proxy address.

28. An unauthorized-email rejection component for use with an email-receiving server for receiving email sent to a user on a network comprising an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, wherein said unauthorized-email rejection component intercepts and rejects any incoming email addressed to the email address of the user if the processing results of the ASL list returns with an "unauthorized sender" result, and further including a redirector module for sending a message inviting the sender of the rejected email to confirm that the sender is an authorized sender of email to the intended recipient by passing an interaction procedure to show that the sender is not a mechanical program attempting to automatically validate the sender, whereupon the validated sender's email address can be added to the ASL list.

29. An unauthorized-email rejection component for use with an email-receiving server for receiving email sent to a user on a network comprising an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, wherein said unauthorized-email rejection component intercepts and rejects any incoming email addressed to the email address of the user if the processing results of the ASL list returns with an "unauthorized sender" result and returns an industry standard "no such user" error code before the email can be accepted for delivery to the user.

30. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user,

wherein the unauthorized-email rejection component is positioned for operation in the flow of incoming email physically or logically upstream before the email-receiving server such that unauthorized email is intercepted and prevented from reaching the email-receiving server.

31. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user,

wherein the unauthorized-email rejection component is positioned for operation in the flow of incoming email

physically or logically upstream before the email-receiving server such that unauthorized email is intercepted and prevented from reaching the email-receiving server, and

wherein, upon the ASL module determining that incoming email has a sender address that is not that of an authorized sender, said unauthorized-email rejection component rejects the incoming email with an industry standard "no such user" error message.

32. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user,

wherein the unauthorized-email rejection component is positioned for operation in the flow of incoming email physically or logically upstream before the email-receiving server such that unauthorized email is intercepted and prevented from reaching the email-receiving server, and

wherein the ASL module includes an ASL list manager for analyzing email header information including FROM and TO addresses of email sent by users to dynamically update the ASL list of authorized senders.

33. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user,

wherein the unauthorized-email rejection component is positioned for operation in the flow of incoming email physically or logically upstream before the email-receiving server such that unauthorized email is intercepted and prevented from reaching the email-receiving server, and

wherein said unauthorized-email rejection component includes a redirector module for sending a message inviting the sender of the rejected email to confirm that the sender is an authorized sender of email to the intended recipient by passing an interaction procedure to show that the sender is not a mechanical program attempting to automatically validate the sender.

34. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user,

wherein said system includes a proxy address module for allowing a user to designate a destination proxy email address for use by a sender in instances where the email address of an authorized sender is not yet known, and if the destination proxy email address is used on incoming email, said unauthorized-email rejection component accepts the incoming email and sends it to the user.

35. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user,

(c) an unauthorized-email rejection component having an authorized senders list (ASL) module which maintains email addresses of senders authorized to send email to the user, wherein the unauthorized-email rejection component is operable with the email-receiving server for intercepting and rejecting any unauthorized email addressed to the email address of the user,

wherein said unauthorized-email rejection component includes a redirector module for sending a message inviting the sender of the rejected email to confirm that the sender is an authorized sender of email to the intended recipient by passing an interaction procedure to show that the sender is not a mechanical program attempting to automatically validate the sender,

wherein the interaction procedure includes a display of a graphic image of a word or object, and a request to the sender to enter a text word in response to the graphic image, whereby the system can confirm that the interaction procedure is not being performed by a mechanical program, and thereupon add the validated sender's email address to the ASL list.

* * * * *

60/180,937

E

A/PROV

02-11-00

Approved for use through ... Patent and Trademark Office. EL529233535US

PROVISIONAL APPLICATION COVER SHEET

Mailed as of Feb. 8, 2000

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53 (b)(2).

Docket Number	PKAY-P1A	Type a plus sign (+) inside this box ->	+

INVENTOR(s)/APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
Katsikas	Peter	L.	Honolulu, HI

TITLE OF THE INVENTION (280 characters max)

UNWANTED EMAIL FILTERING SYSTEM

CORRESPONDENCE ADDRESS

Leighton K. Chong
Ostrager Chong Flaherty & Onofrio
841 Bishop Street, Honolulu, HI 96813-3908

STATE	HI	ZIP CODE	96813-3908	COUNTRY	US

ENCLOSED APPLICATION PARTS (check all that apply)

- [X] Specification — *Number of Pages* 9
- [X] Drawing(s) — *Number of Sheets* 8
- [X] Small Entity Statement
- [ ] Other (specify)

METHOD OF PAYMENT (check one)

- [X] A check or money order is enclosed to cover the Provisional filing fees
- [ ] The Commissioner is hereby authorized to charge filing fees and credit Deposit Account Number:

PROVISIONAL FILING FEE AMOUNT ($) $ 75.0

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

- [X] No.
- [ ] Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE [signature] Date 02/08/00

TYPED or PRINTED NAME Leighton K. Chong REGISTRATION NO. *(if appropriate)* 27,621

- [ ] Additional inventors are being named on separately numbered sheets attached hereto

PROVISIONAL APPLICATION FILING ONLY

PTO/SB/09 (12-97)
Approved for use through 09/30/00. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(b))—INDEPENDENT INVENTOR	Docket Number (Optional) PKAY-P1A

Applicant, Patentee, or Identifier: Peter L. Katsikas

Application or Patent No.:

Filed or Issued:

Title: UNWANTED EMAIL FILTERING SYSTEM

As a below named inventor, I hereby state that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees to the Patent and Trademark Office described in:

[X] the specification filed herewith with title as listed above.

[ ] the application identified above.

[ ] the patent identified above.

I have not assigned, granted, conveyed, or licensed, and am under no obligation under contract or law to assign, grant, convey, or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern, or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

[X] No such person, concern, or organization exists.

[ ] Each such person, concern, or organization is listed below.

Separate statements are required from each named person, concern, or organization having rights to the invention stating their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

NAME OF INVENTOR	NAME OF INVENTOR	NAME OF INVENTOR
Peter L. Katsikas		
Signature of Inventor	Signature of Inventor	Signature of Inventor
8/30/1999		
Date	Date	Date

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Specification for SpamKapu

(a) Application transmittal form.

(b) Fee transmittal form.

(c) Title of the Invention.

"SpamKapu" Software to eliminate unauthorized receipt of electronic mail (spam)

(d) Cross Reference to related applications (if any).

Internet SMTP, POP3, and related standards

(e) Statement of federally sponsored research/development (if any).

none

(f) Reference to a microfiche appendix (if any).

none

(g) Background of the Invention.

Not sure here.

(h) Brief Summary of the Invention.

Most, if not all, of the current software to control spam is based on identifying lists of spam sources or senders and filtering email based on those lists. This technology is only as good as the identifying list and cannot guarantee that the user will not receive spam. Today's spam control software assumes all email is authorized an attempts to filter out unauthorized email.

Because today's spam filtering technologies are based on lists of known spam sources, it is impossible for them to filter email that comes from non-SPAMMERs that is still undesired by the user. For instance, one may have disclosed their email address at a Web site which now used by individuals that are sending email to the user. These individuals will never appear of spam lists because technically they are not spamming.

SpamKapu based on the idea that all email is unauthorized and must be compared against an "authorized senders" list in order to be acceptable to the user. This filters not only spamming sources, but <u>any</u> sender which the user deems as unauthorized. This creates an inherently powerful and 100% private email solution.

SpamKapu intelligently formulates the "authorized senders" list based on analysis of the user's email usage (such as sent email) and a gathering of key data such as their known contacts and associates. The authorized senders list may also be easily

manipulated by the user at any time to add or remove authorized senders.

To summarize: SpamKapu effectively blocks 100% of unauthorized email to the user. It is based on the idea that if you did not send someone an email, they are not authorized to send you email.

(i) Brief description of the several views of the drawing (if any).

(j) Detailed Description of the Invention.

SpamKapu is formed of several key modules and definitions:

SUBSCRIBER: the person that using SpamKapu.

FRIEND: an email-sending source that is authorized to send email to the SUBSCRIBER.

SPAMMER: an email-sending source using manual or highly mechanized means to send one or more emails through the Internet that is not authorized to send email to the SUBSCRIBER.

CONTACT: an email-sending source that is a human being attempting to reach the SUBSCRIBER for a legitimate cause.

SUSPECT: an email sending source that has not yet been identified as either a SPAMMER or CONTACT.

ASL Manager

Software designed to populate the ASL from a variety of methods:

Contact lists of the user indicating Friends.

Continual analysis of sent mail logs which may expose additional Friends.

Standard file formats (i.e. comma-delimited) which would allow subscribers to easily update their ASLs.

Spam Processor (SP)

Decides whether an email address is FRIEND, or SPAMMER by executing rules on the SPDB in conjunction with the ASL.

Returns this result along with any message to include in the error response to the REDIRECTOR.

Uses industry standard PERL programming syntax and incorporates as PERL interpreter to execute rules.

Spam Processing database (SPDB) of which a unique copy exists for each SUBSCRIBER, composed of several tables:

Authorized sender list (ASL), containing

An email address or matching pattern for an email address

Default: exact match

A specific email address

john@company.com

UNIX Standard wildcard matching

*.microsoft.com i.e. anything from "Microsoft.com"
microsoft: anything with microsoft in it
*.mil: any email from the military

Matching any known "blackhole list" by using a %BLACKHOLE% symbol.

A conditional and parameters to execute if the match is true
An action and parameters to perform if the conditional is true.
A parameter used by the secondary action
The last date the SUBSCRIBER sent email to this address
The last date this address sent email to the SUBSCRIBER
Date the record was created

Example list of conditionals to be used by the SPAM PROCESSOR, e.g:

expiration dates.

A given address until 2/12/2004

Date ranges

A given address from 4/1/2004 to 5/2/2004

Specific recurring times

first week of every month but no other time.

e.g newsletter@magazine.com
acceptable during 1st week of each month.

A link to external software designed to allow for additional user-defined criteria

This allows for 3rd party applications.

Example list of different secondary actions to take

Send a given message in the error response.
Send a given message as an email.
Open a file and email its contents
Open a file and send its contents as an error reponse.
Set the sender's status to SPAMMER or FRIEND
Give SMTP default error message
Link and execute external software designed to allow for additional user-defined actions

This allows for 3rd party applications.

List of messages that may be invoked by a given secondary action

Standard "error"
Custom with variable substitution in the message body, e.g:

%username% is substituted with the sender's email address
%subid% is the ID code of the subscriber
%date% is today's date

"hello %username% you have been identified as spam, go to http://www.spamkapu.com/subscriber=%subid% and if you're really human we'll let you in.

Custom text: "All email addresses from America Online are unconditionally rejected"

Authorized Sender mailbox (ASM)

An electronic mailbox confirming to popular Internet standards (as of this writing, POP3 & IMAP4) that contains email sent from FRIENDS.

SpamKapu email address (SKE)

An Internet SMTP-complied email address provided by spamkapu that is unique to the SUBSCRIBER.

Redirector

Software that intercepts incoming email sent to the SKE, routes it's sender's email address to the SP for validation (FRIEND, or SPAMMER)

If FRIEND, the email is directed to the ASM.

If SPAMMER, the SPAMMER is given an error message similar to one if the user didn't exist along with information on how to access the SUBSCRIBER's WBM should further communication be desired.

Web-based messenger (WBM)

A Web site that designed to determine if a SUSPECT is either a SPAMMER or a CONTACT.

An online form would be presented to the SUSPECT to allow entry of the intended message to the SUBSCRIBER.

This form would operate in such a way that only a human SUSPECT would be able to properly execute the form.

A unique web page with a random word would be generated

The SUSPECT would be prompted to enter the word.

If the word matched, the form would be considered "operated by a human" and the SUSPECT is now deemed as a CONTACT

If validated as a CONTACT, the message in the form along with the CONTACT's email address would be sent as a special email to the SUBSCRIBER's ASM. The subject line of the email would contain the word "contact:" so it could easily be filtered or be subject to special processing by an industry standard email client.

The SUBSCRIBER would have the opportunity to read the email, knowing that at least it was sent by a CONTACT.

ASL manager

Software that intercepts all sent email from the SUBSCRIBER and copies the recipients along with other information into the ASL

- The ASL manager may work on either a dynamic (as emails are sent) or batch (analyze logs or other data sources).
- The ASL works in conjunction with the UMM

- SMTP manager
  - Software that provides a SUBSCRIBER with an SKE and interfaces with other Internet SMTP standard functions such as:
    - Sending email FROM or TO the subscriber through the ASL manager.
    - For example, the SMTP manager may interface with the subscribers "official" or known corporate address to eliminate spam sent to the corporate email system.
- User maintenance modules (UMM)
  - A set of software utilities that allow the SUBSCRIBER to maintain personal settings and the ASL. Examples include:
  - Default expiration settings.
  - Bulk-loading of friends into the ASL
  - Search/add/edit/delete ASL entries
  - Handling of mail once sent to PSM (i.e. create a predefined response to the spammer)
  - Outright rejection of email, disallowing it to even get to the PSM.
  - Preview/Delete items from the PSM
  - Other features of benefit to subscribers

SpamKapu may be packaged in a variety of ways

- As an online service (i.e. Web site) that allows users to subscribe and realize the benefits of spam-free email services.
- As a server-side software package. By installing SpamKapu at the server level, any/all users with a valid account on the server can receive the benefits of spam-free email. This is an idea solution for ISPs and/or larger organizations with their own server resources.
- As a client-side software package. SpamKapu can install on popular email clients such as Outlook and provide near-identical functionality. This is an idea situation where the user's server does not have SpamKapu installed

Operation of the Invention

- As a server-side software package or online service
  - SUBSCRIBERs are added to SpamKapu system.
  - Each SUBSCRIBER is provided with a PSM, ASM, and UMM and an SKE.
  - The SUBSCRIBER changes appropriate setting on their email software to accomplish the following:
    - Use current Internet standards (currently POP3 or IMAP4) to retrieve mail from both the PSM and ASM

Redirect email sent to their current email address to their SKE instead OR set the email reply-to address to their SKE.
Use the SMTP manager to handle the sending of all email.

Any email sent to the SKE is processed by the redirector as described above.

Any email sent by the SUBSCRIBER through the ASL manager (via the SMTP manager) and processed as described above.

The user can retrieve email from the ASM at any time using Internet standards (currently POP3 or IMAP4).

The user can retrieve email from the PSM at any time using Internet standards (currently POP3 or IMAP4) user other software that can delete, further filter, or altogether discard the contents.

SUBSCRIBERs may interact with the UMM at any time.

Use of SMTP-standard email error response codes as a matter of rejecting user-specific spam

This is being used today, but only where a given email server is rejecting ALL email from a given NETWORK.
This claim is against SPECIFIC email directed to a SUBSCRIBER that is identified to have originated from a A SPAMMER.

As a client-side software package on an Outlook 98 or greater client.

After installation, a folder labeled "PSM" will be created.

Users may interact with the a client-side installation of the UMM at any time.

ASM will be sent to the standard "inbox" folder and PSM will be sent to the "PSM" folder.

Other operations are similar to the server-side package or online service described above.

(k) Claim or claims.

Any software which analyzes the user's personal email usage patterns to create an ASL or equivalent and in-turn uses this ASL to make decisions on how to process incoming email.

Analysis of sent email and received to determine and refine the ASL.

Analysis and rejection of SPAM at the lowest (earliest) possible level in the mail transmission protocol such that SPAMMERS receive error messages indicating the user doesn't even exist.

SUBSCRIBER never even processes or downloads email.

Analysis of contact databases to determine and refine the ASL.

Analysis of email logs (both sent and received) to determine and refine the ASL.

Methods to only allow humans to access the WBM to send messages to the SUBSCRIBER.

Methods for 3rd party software to interface with SPAMKAPU to broaden the scope and functionality of determining if email is SPAM or not, for example:

Analysis against 3rd party databases such as the Better Business Bureau

Methods for 3rd party software to interface with SPAMKAPU to broaden the scope and functionality of the various types of actions that can be taken on SPAM.

Methods for 3rd party software to interface with SPAMKAPU to broaden the scope and functionality of analyzing power

(l) Abstract of the disclosure.

(m) Drawings (if any).

(n) Executed oath or declaration.

(o) Sequence listing (if any).

(p) Plant Color Coding Sheet (applicable in plant patent applications).

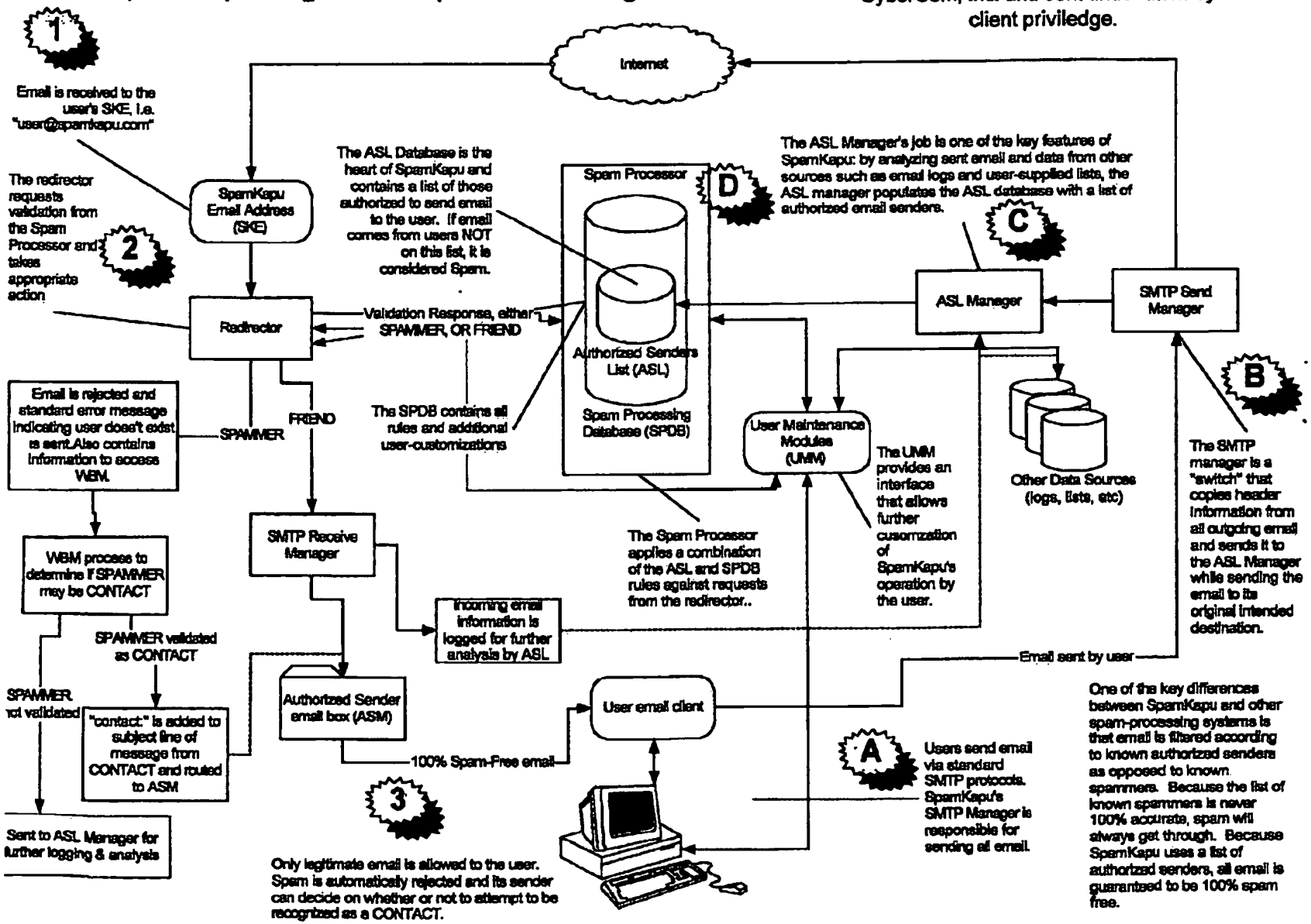
Other

dataflow of overall system
server-based architecture of system
client-based architecture of system
list of other ideas and uses of the system and variances to do the same thing (other than spamming)

references to other technology used

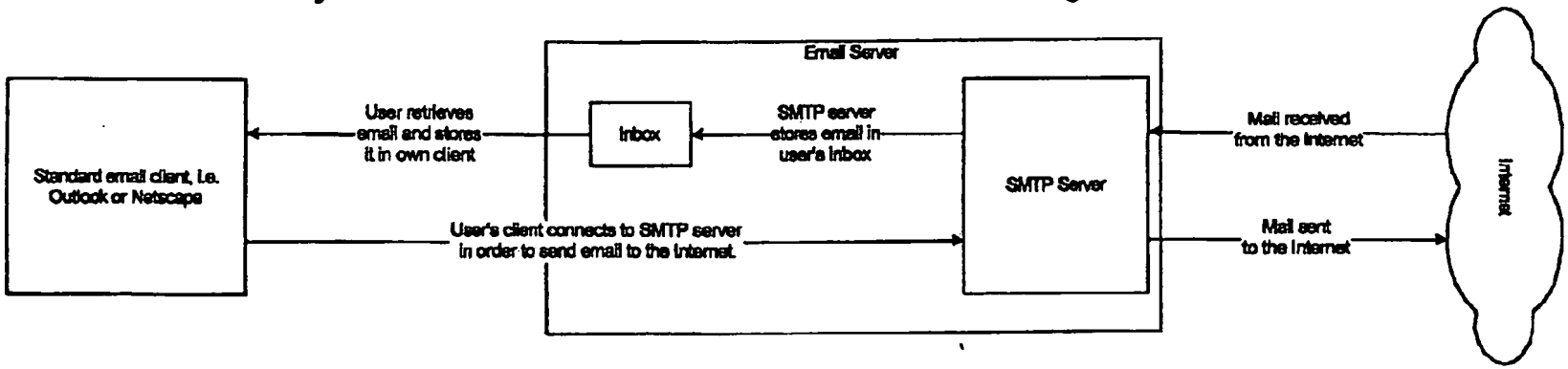
RFC 821
SMTP email
PERL
Sendmail
RealTime Blackhole List (www.mail-abuse.org)

"SpamKapu" high-level operational diagram.



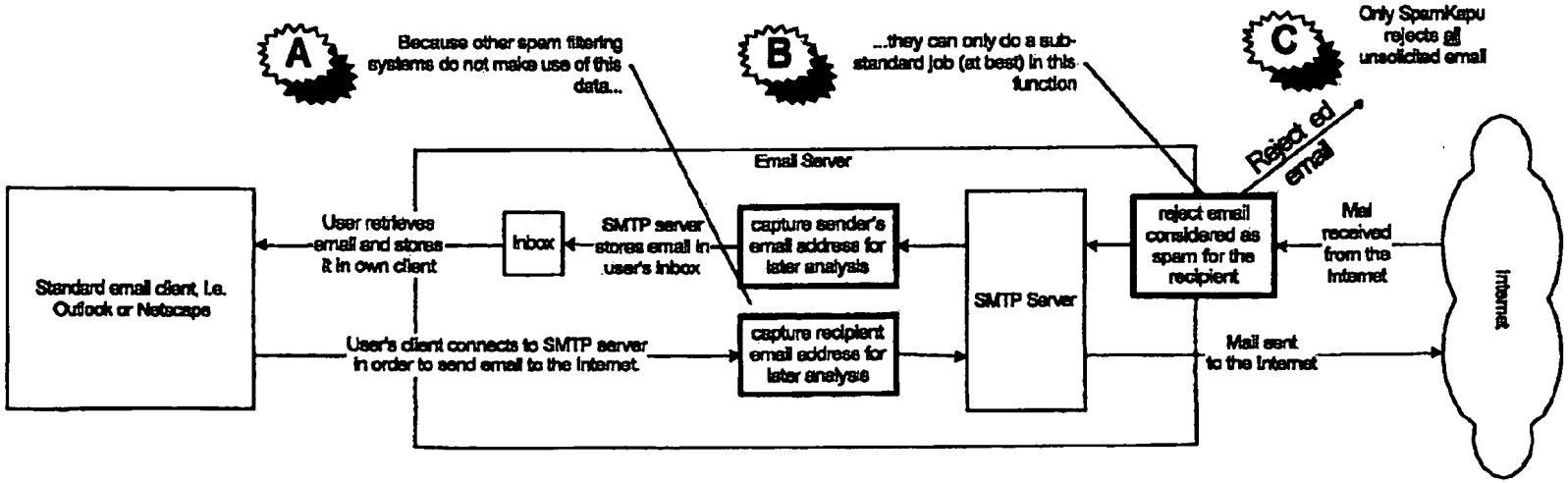
"SpamKapu" Conceptual Organization

Today's "Standard" SMTP send/receive block diagram and data flow



Spamkapu additions to SMTP send/receive block diagram and data flow

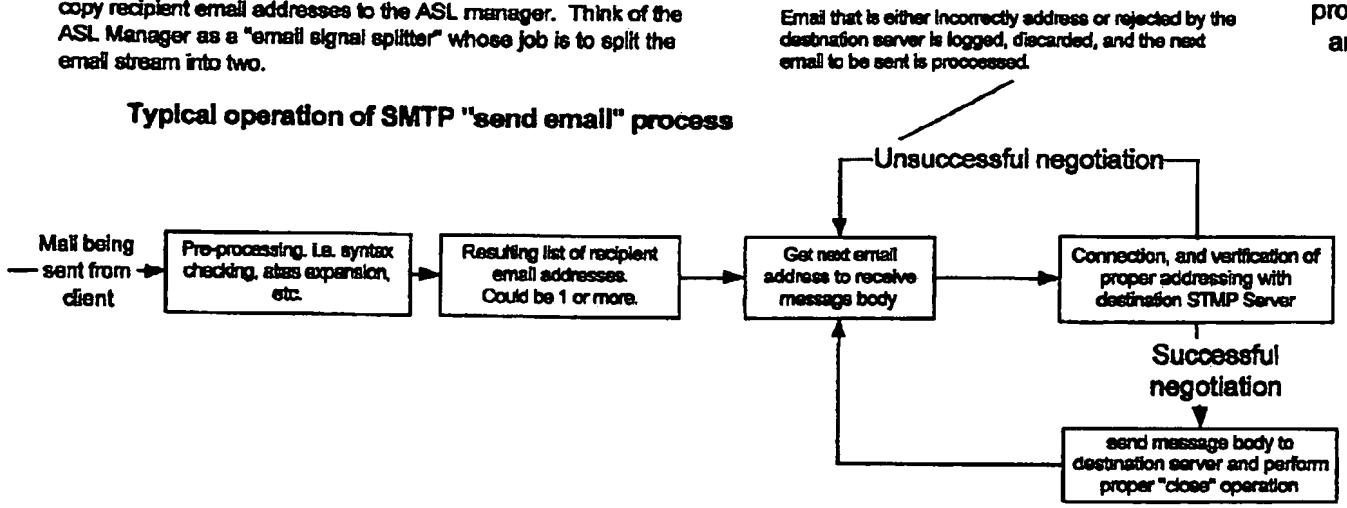
Additions shown in **bold** blocks



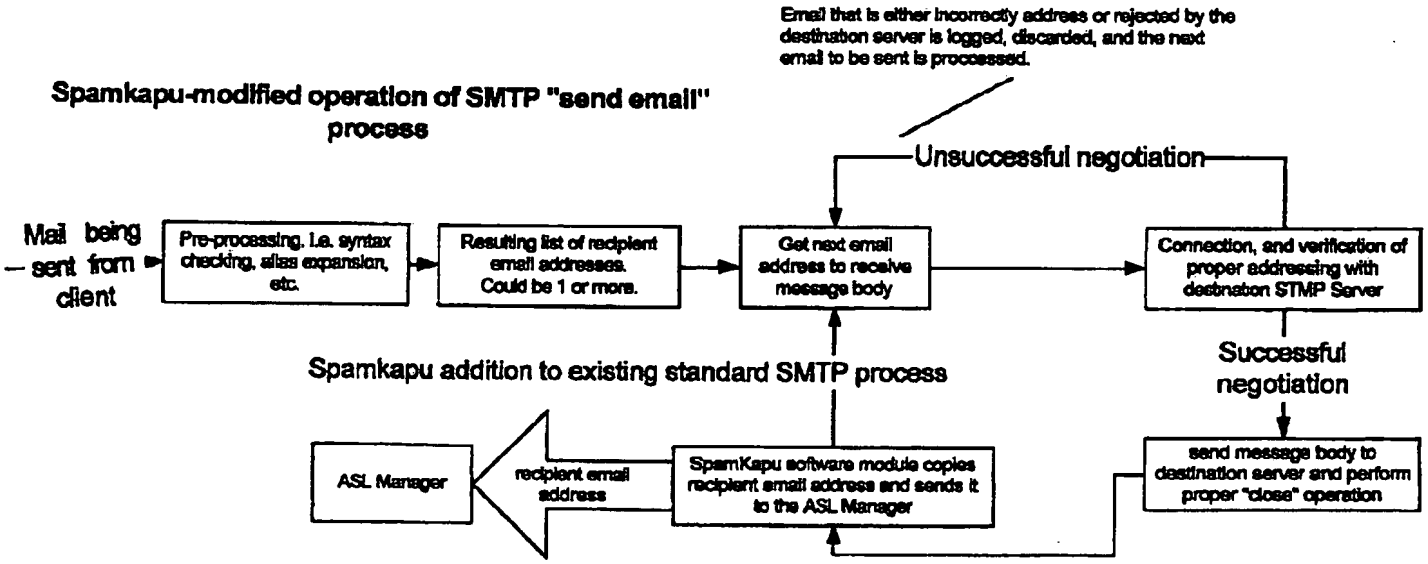
"SpamKapu" detailed operation of SMTP Manager

The STMP Manager's key role is to intercept outgoing email and copy recipient email addresses to the ASL manager. Think of the ASL Manager as a "email signal splitter" whose job is to split the email stream into two.

Typical operation of SMTP "send email" process

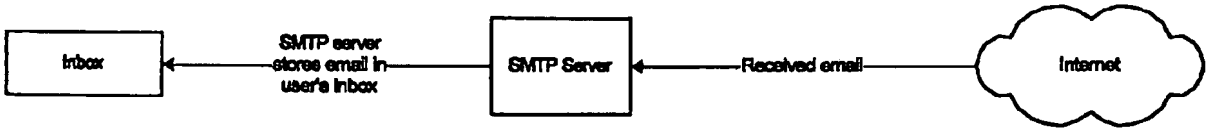


Spamkapu-modified operation of SMTP "send email" process



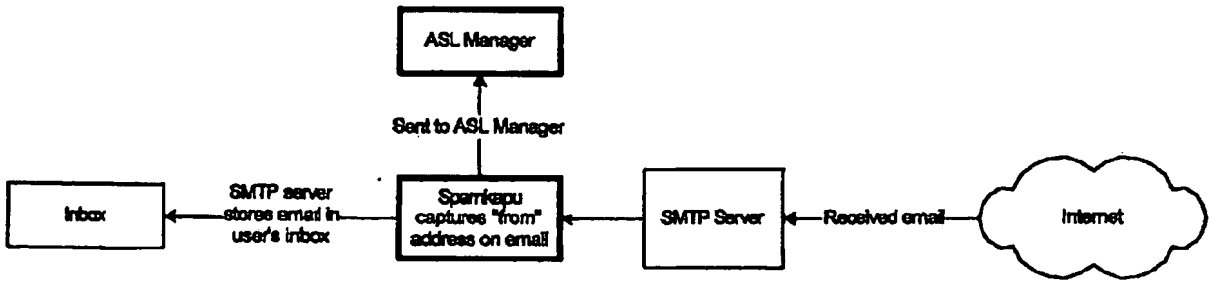
"SpamKapu" detailed operation of SMTP Receive Manager

Today's "Standard" SMTP receive block diagram and data flow

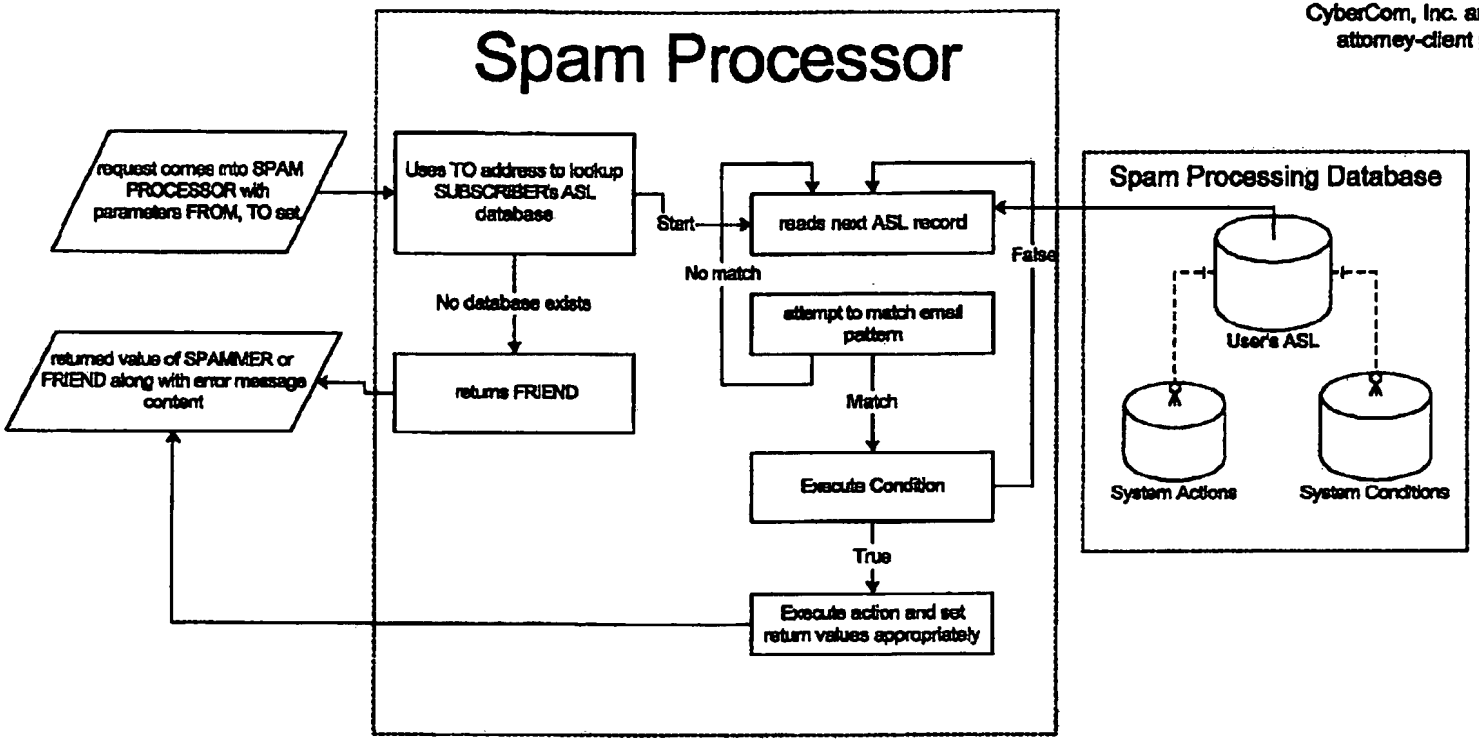


Spamkapu additions to SMTP send/receive block diagram and data flow

Additions shown in **bold** blocks



"SpamKapu" detailed operation of SPAM PROCESSOR



"SpamKapu" detailed operation of SPAM PROCESSOR DATABASE

1. Using FROM email address supplied, loop through each line of the ASL until there is a pattern match in this field

2. Once there is a pattern match, execute the condition based on the code in this field

3. If the condition returns TRUE, execute the action based on the code in this field and exit. If condition returns FALSE, continue looping through each email pattern

Get Condition — Get Action — Execute

ASL Table (example)

email pattern	condition	action if true	Comment
mom@gte.net	always	friend	this is my mom so she's always a friend
sales@acme.com	activerange (2/5/2004, 6/5/2004)	friend	I'm working with this businessperson only from Feb 5 to June 5
john@home.com	before (12/1/2003)	friend	After 12/1/2003, I do not want messages from this address
*@aol.com	always	email (SPAMMER, "noaol.txt")	email we get from aol gets send back with an explanation
*	blackhole()	emailtmother (SPAMMER, blacklist.txt, abuse@%domain% %SENDER%, admin@spamkapu.com)	run a 3rd party software, mark as a spammer, send a standard message to the system's administrator, and mark it as coming from spamkapu's administrator
*	activerange (3/14/*, 3/20/*)	friend	my birthday falls on 3/17, so from 3/14 to 3/20, I'll let anyone pass through to wish me a happy birthday.
*	always	errormsge(spammer, contact.txt, 550)	mark as spammer, send the "you might be a contact" message, return "No such user" error code

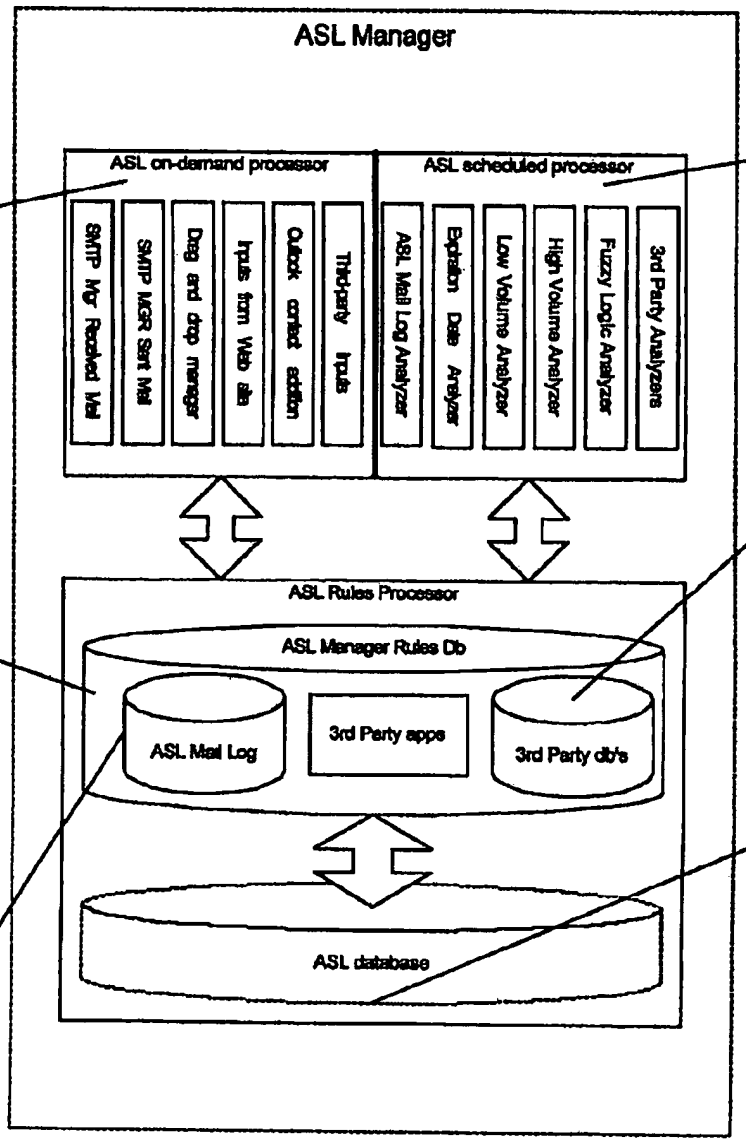
Condition Table (example)

condition	syntax	description	code
always	always	return true always	return(true)
activerange	activerange (low date, high date)	true if today is within a range	if today >= %1 and today <=%2 then return(true)
runcode	runcode(filename)	runs long and lengthy code program	exec(file.pl)
blackhole	blackhole()	3rd party software to check against a blackhole list	exec(blackhole.exe)
before	before(date)	true if today is less than date	if today <= %1 then return(true)

Action Table (example)

action id	syntax	description
email	email(ident, file)	return parm1 as identifier, get 2nd parameter as a filename, translate and email it.
errormsg	errormsge(ident, file, errorcode)	return parm1 as identifier, get 2nd parameter as a filename, translate and return it as the error code
external	exec(file)	execute 2nd parameter as an external 3rd party software package and return its result as the
emailrtn	emailtmother (ident, filename, to, from)	return parm1 as identifier, get 2nd parameter as filename of message text, send it as an email to 4th parameter, and use 3rd parameter as FROM
friend	return(SPAMMER)	return value as FRIEND, no parms needed
spammer	return(SPAMMER)	return value as SPAMMER, no parms needed

"SpamKapu" detailed operation of ASL Manager



Content arrives into the ASL "live" from a variety of sources and easily incorporates 3rd party plugins.

The classic example is of course are email addresses being sent to and received from the SMTP Manager, however it doesn't have to stop there. There are a wide variety of other sources from which email addresses may be gleaned.

After content has arrived either on-demand or via scheduling, the rules processor determines how to add, update, or modify the ASL database

The ASL Mail log can be populated by any on-demand or scheduled process. It contains a very rich history of the SUBSCRIBERS available for analysis.

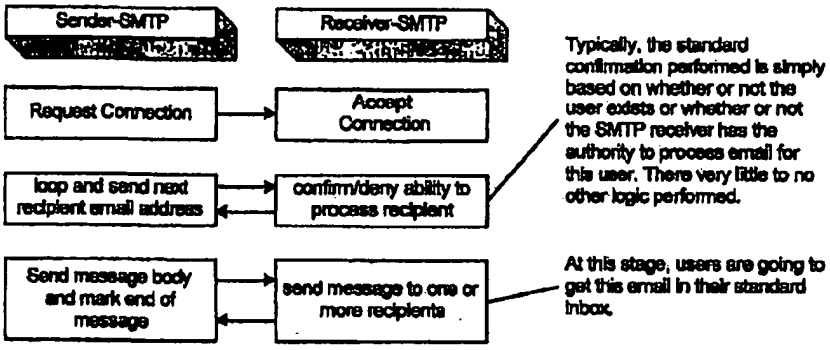
The ASL Manager also runs tasks on a scheduled basis for analysis and maintenance functions. This allows a very rich examination of the SUBSCRIBER's ASL database and mail log to continually refine the database accuracy and relevance.

The system's architecture allows for easy integration of 3rd party solutions so that SpamKapu can harness the collective power of the industry to continually extend and improve its feature set.

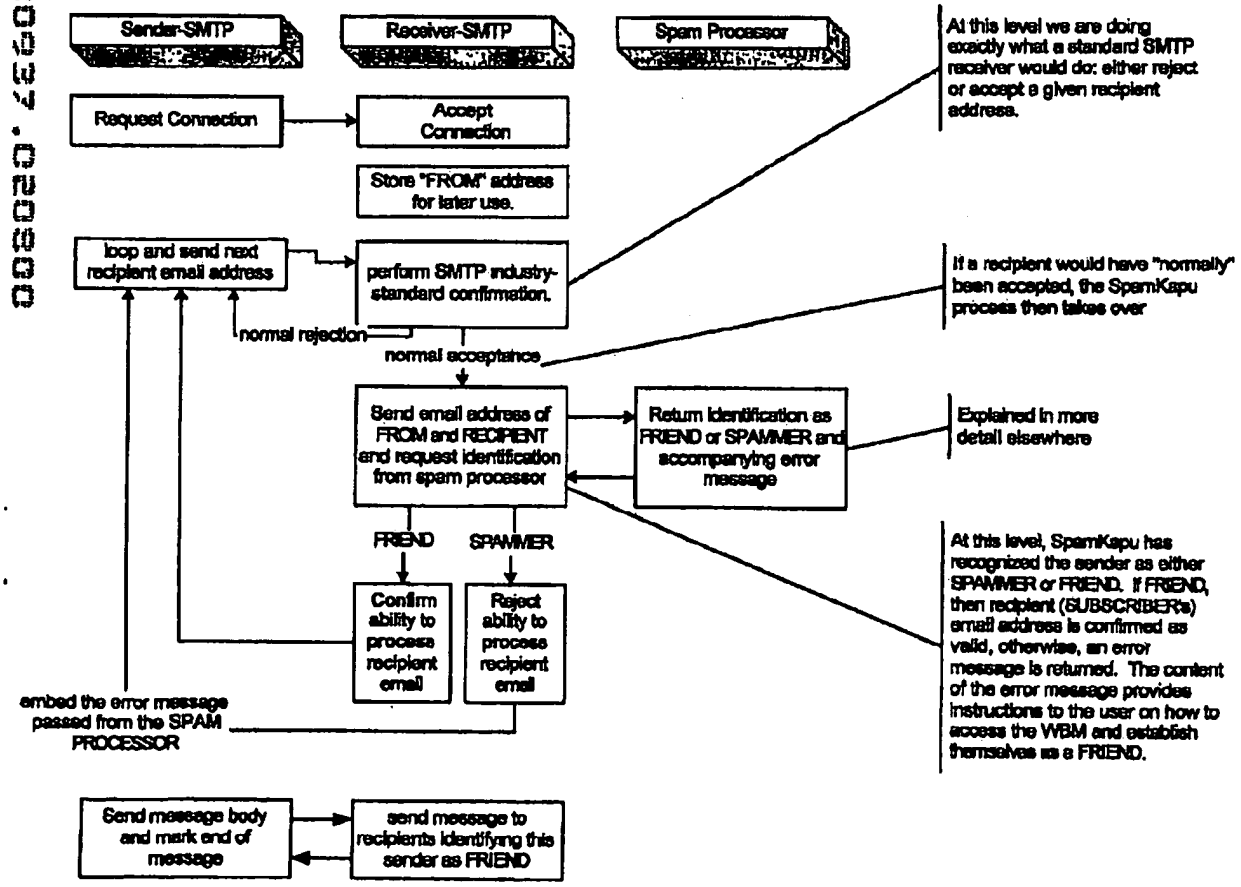
The ultimate result of this architecture is to create a very richly detailed ASL database which goes beyond total elimination of spam by continually reflecting the current needs of the SUSSCRIBER dynamic use of email

"SpamKapu" detailed operation of REDIRECTOR

Typical operation of SMTP "send email" process



SpamKapu-modified operation of RECEIVER-SMTP process, known as the REDIRECTOR



09/648,894
8/25/2000

F

SYSTEM FOR ELIMINATING UNAUTHORIZED ELECTRONIC MAIL

This U.S. patent application claims the priority of U.S. Provisional Application 60/150,025, filed on September 1, 1999, entitled "Unwanted Email Filtering System", and U.S. Provisional Application 60/180,937, filed on February 8, 2000, entitled "Unwanted Email Filtering System", both by the same inventor.

FIELD OF THE INVENTION

This invention relates to a system for eliminating unwanted email, and particularly to one in which all email must be recognized as sent by an authorized sender in order to be accepted.

BACKGROUND OF THE INVENTION

Unwanted or unauthorized email is a significant bane for users on worldwide networks, such as the current public Internet. Once a person's email address becomes known in a network system, it can easily be replicated in computerized lists and passed on electronically to an unlimited number of parties who have not been authorized or invited to send email to the user. A user's electronic mailbox can become inundated with such unauthorized email. Unauthorized or unwanted email is referred to generically in the industry by the term "spam", although the term is not intended to be associated with or to disparage the popular canned meat product sold under the trademark "Spam" by Hormel Corp. The user may have an email address with a commercial information service provider (ISP) service which limits the amount of email that can be accepted and/or stored or which charges the user by the volume received. The user may also waste a significant amount of time opening and reviewing such unwanted email. Unauthorized email may also be sent by unscrupulous persons who may enclose a virus or noxious software agent in the

email which can infect the user's computer system, or which can be used as an unauthorized point of entry into a local network system that handles the user's email.

Most, if not all, of the current software to control the receipt of spam is based upon the use of identifying lists of known spam sources or senders ("spammers"). Such conventional spam control software functions on the basis of receiving all email as authorized unless a sender is identified as being on the exclusion list and the email can be filtered out. This approach is only as good as the identifying list and cannot guarantee that the user will not receive spam. Spammer lists require frequent updating and must be distributed in a timely manner to all subscribers to the spam control software or service. Sophisticated spammers frequently change their source Internet address, and can defeat attempts to keep exclusion lists current. They can also route the unwanted email through the Internet servers of other parties so as to disguise the source of the emails through innocuous or popularly recognized names. A user's email address may also become known to large numbers of individuals in public chat rooms or on public bulletin boards. Unwanted email sent by individuals are not tracked on spammer lists, because the sending of email by individuals is technically not spamming.

SUMMARY OF THE INVENTION

Accordingly, it is a principal object of the present invention to provide a spam control system that cannot be defeated by spammers who frequently change their source addresses or disguise themselves by routing email through other servers, or by individuals who send email that are not invited or authorized by the user. It is a particular object of the invention that the system of the invention reject all email as unauthorized unless the sender is recognized as being on the user's acceptance list.

In accordance with the present invention, a system for eliminating unauthorized email sent to a user on a network comprises:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user, said email-receiving server having an authorized senders list (ASL) module which maintains an ASL list of email addresses of external users authorized to send email to the user, and

(c) an email rejection module operable with the ASL module for rejecting the receipt of email sent to the email address of the user if the email address of the sender is not one that is maintained on the ASL list for the user.

In a preferred embodiment, the system's ASL module includes an ASL database for storing ASL lists of authorized sender addresses for respective subscribers of the system, a spam processor module for checking the ASL lists for matches, and an ASL manager for creating, maintaining, and updating the ASL lists. A redirector module rejects email if, upon sending a request for validation to the spam processor module, the sender's address does not match any authorized sender address found on the ASL list. Email rejected by the redirector module is redirected to a web-based messaging (WBM) module which sends a message notifying the sender to confirm that the sender is a legitimate sender of email to the intended recipient. If the sender logs on to confirm their status, the WBM module executes an interaction procedure which can only be performed by a human, in order to ensure that the confirmation procedure is not performed by a mechanical program. The ASL manager maintains the ASL lists based upon sender address data collected from various sources and analyses of various email usage factors, including sent email, received email, contact lists maintained by the user, user preference inputs, third party programs,etc.

The invention also encompasses associated methods of performing the above functions, as well as related software components which enable these functions to be performed.

Other objects, features, and advantages of the present invention will be described in further detail below, with reference to the following drawings:

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1A is a block diagram illustrating a standard Internet email system using the conventional method for filtering email from spammers (Prior Art), as compared to **FIG. 1B** which shows a conceptual overview of a system in accordance with the present invention.

FIG. 2 is a process flow diagram for a preferred embodiment of the anti-spam system of the present invention.

FIG. 3A is a block diagram illustrating a standard SMTP send email process (Prior Art), as compared to **FIG. 3B** which shows a modified send email process used in the present invention.

FIG. 4A is a block diagram illustrating a standard SMTP receive email process (Prior Art), as compared to **FIG. 4B** which shows a modified receive email process used in the present invention.

FIG. 5 is a process flow diagram illustrating the operation of an anti-spam processing routine in the preferred embodiment of the invention.

FIG. 6 is a process flow diagram illustrating the detailed operation of a Web-Based Messenger (WBM) routine for handling email initially rejected by the anti-spam control.

FIG. 7A is a block diagram illustrating a standard SMTP send-receive email handling process (Prior Art), as compared to **FIG. 7B** which shows a modified Redirector process for handling received email.

FIG. 8 is a schematic diagram illustrating the structure and operation of the ASL Manager in the preferred embodiment of the spam control system.

FIG. 9 illustrates a detailed implementation of examples of processing of email send/receive and user contact data into specific forms of actions taken by the ASL Manager.

DETAILED DESCRIPTION OF INVENTION

In contrast to the known approaches of existing spam control methods of accepting all email unless listed on an exclusion list as unauthorized, the fundamental principle of the present invention is to reject all email unless listed on an inclusion list as authorized. In this manner, it is possible to filter out email that comes from unrecognized spammers as well as individuals who send email that is uninvited by the user. Unlike the known email filtering systems, the present invention does not attempt to filter out the unwanted email after it has been accepted. Rather, it outright rejects the email at the earliest entry level. Thus, the invention operates on the premise that all email will be treated as unauthorized unless the sender is found to be on an "authorized senders" list in order to be accepted by the user. This provides an inherently powerful and 100% effective spam control solution in an environment where spammers can instantaneously change their source address or apparent identity and individuals in public areas can obtain email addresses of other users and send them unwanted email.

The following is a detailed description of one preferred embodiment of a system for implementing the invention concept. In this embodiment, the spam control system intelligently formulates the "authorized senders" list based upon an ongoing analysis of the user's email usage, such as to whom and with what frequency sent email is addressed to other users, and through the gathering of high-level user contact data, such as a user's known contacts and associates identified on other lists or files maintained by the user which indicate persons considered as authorized. The "authorized senders" list may also be updated and manipulated by the user at any time to add or

remove authorized senders. While this specific implementation is used, and certain components are provided and configured to be interoperable in the described ways, it is to be understood that the full scope of the invention is deemed to encompass many other suitable modifications and variations to the described guiding principles of the invention.

FIG. 1A is a block diagram of a standard email system for sending and receiving email on the Internet and is used to explain the conventional method for filtering out email from spammers. The system follows a standard industry protocol for handling email on the Internet, referred to as SMTP. Users typically subscribe with a chosen ISP for Internet access and related services, including email services. The users access the Internet through the ISP using a dialup or high-speed line connection and a standard browser. The browser includes or functions with a standard email client 101, such as the Outlook ™ email client distributed by Microsoft Corp., headquartered in Bellevue, Washington, or the Netscape ™ email client used by AOL/Netscape, headquartered in Fairfax, Virginia. The ISP operates at a website address corresponding to its domain name which is addressable by users on the Internet. The ISP's service functions are performed for a large number of subscribers through one or more servers. Typically, an email server 102 is used to handle the email service functions. Email sent to the ISP from the Internet is received at SMTP Server 102b, where various administrative functions are performed, such as checking whether the addressee is an authorized subscriber of the ISP, then the email is placed in a storage space reserved for that user, referred to as Inbox 102a. When users connect to the ISP, they can retrieve their email and store it with their own email client (on their own computer). Users can send email by composing it locally at their email client, then uploading it to the SMTP Server 102b at the ISP, which then routes it to the recipient's email address on the Internet.

Conventional anti-spam control can be implemented with the SMTP Server and/or at the email client. Many ISPs implement an exclusion list of known spammers at the SMTP Server. In addition, they commonly allow a user to filter out unwanted email from certain senders known to the user. For example, the user's email client may have a filtering function that allows the user to input unwanted sender email addresses to the SMTP Server so that email received by the SMTP Server can be filtered out before being put into the user's Inbox. Further, independent software vendors sell sophisticated email handling programs that work with the user's email client. For

example, some handling program have functions for categorizing received email into topical file folders, and email from unrecognized senders may be put into a "Miscellaneous" or "Unrecognized" file folder.

In FIG. 1B, a conceptual overview of a system in accordance with the present invention is shown. As before, the standard email client 101 is connected to an email server 104 for sending and receiving email to and from the Internet via SMTP Server 104b and Inbox 104a. However, in this modified email server 104, an Authorized Sender List (ASL) Manager captures recipient email addresses from email sent by the user, as shown at block 105, and also captures sender email addresses from email sent to the user, as shown at block 106. The ASL Manager analyzes the captured sender email addresses and recipient email addresses and employs certain pre-defined rules (described in further detail below) to add or remove email addresses from the "authorized senders" list, referred to as the ASL List or Database. The ASL List is used by the SMTP Server 104b to accept only email from senders on the ASL List and place the accepted email in the user's Inbox 104a, while rejecting all other email as "unauthorized", as indicated at block 107.

Referring to FIG. 2, the process flow for the operational steps of the anti-spam system of the present invention will now be described. Certain terms used in the description are defined below:

SPAMKAPU: An example of the spam control system of the invention.

SUBSCRIBER: A person subscribing to an ISP email service that is using the spam control system of the invention.

FRIEND: An email-sending source that is authorized by the spam control system to send email to the SUBSCRIBER.

SPAMMER: An email-sending source that is not authorized to send email to the SUBSCRIBER, which is commonly understood to be an unknown or unauthorized party that is using a manual or computerized email list mailing program to send large volumes of emails repetitively through the Internet.

Potentially
RC-8/24/00

CONTACT: An email-sending source that has been identified by the system as a legitimate correspondent of the SUBSCRIBER is authorized by the system to send email to the SUBSCRIBER.

SUSPECT: An email sending source that has not yet been identified as either a SPAMMER or a CONTACT.

Email sent from the Internet (103) is sent to the email address of the ISP for the SUBSCRIBER, referred to in block 201 as the SpamKapu Email Address (SKE). Received email must first pass through the Redirector 202. The Redirector 202 sends a request for validation for the email from the Spam Processor 203 which maintains the Spam Processing Database (SPDB) 203a, including the Authorized Senders List (ASL) 203b. The SPDB Database and ASL List are the heart of SPAMKAPU, as they contain the lists of persons authorized to send email to the respective SUBSCRIBERS of the system. The Spam Processor 203 sends a response, either that the sender's address on the email is not authorized on the ASL List, i.e., is a SPAMMER, or is authorized on the ASL List, i.e., is a FRIEND. If the response is that it is a SPAMMER, the Redirector 202 rejects the email, as shown at block 204, such as by sending a standard error message to the sending server that the user as addressed does not exist.

As a refinement to the system, a Web-Based Messenger (WBM) process at block 205 may be set up to provide a corrective procedure in the event that the rejected email is from someone not authorized but not listed permanently on the ASL List as a SPAMMER. The unauthorized email may actually be from a person who has not been previously processed in the anti-spam system but who has a legitimate reason to reach the SUBSCRIBER. The WBM process 205 is set up as part of the spam control system to which the rejected email is redirected. Upon receipt of the redirected email, the WBM process stores it in the WBM database, assigning the email a unique ID code and also an expiration date. The WBM process then sends an error response email to the email sender, who is now treated as a SUSPECT. For example, the error message may read:

> "An email sent by you to SUBSCRIBER's address was redirected to this site as being sent from an unrecognized sender address which may be a source of spam email. If you would like to confirm yourself as a person with legitimate reason to

reach the SUBSCRIBER, please visit the WBM site (or send a reply email) and confirm your status as a CONTACT."

The WBM may have a separate web site address for interactions with SUSPECTS, or it may be set up to receive and recognize email responses from SUSPECTS. When a SUSPECT receives the error response email, if they are a legitimate CONTACT for the SUBSCRIBER, they may elect to go to the WBM site or send a reply email in order to confirm their status as a legitimate CONTACT. If done before the expiration date, the WBM process will add a special codeword such as "contact:" to the subject line of the redirected email, as shown at block 206, and re-route the email to the Authorized Sender Mailbox (ASM) 209. The sender address for email re-directed through this process is also stored (as indicated by the dashed line to block 210) and logged for further analysis by the ASL Manager 211, to determine if the status of the SUSPECT should be upgraded to FRIEND and added to the ASL 203b. If the SUSPECT does not respond, this fact is also sent to the ASL Manager for further analysis. The extra confirmation step effectively eliminates SPAMMERS since they use automated programs to send out batch email and typically will not take human response time to log on to the WBM site or send a reply email to confirm their legitimate status.

If the Spam Processor sends a validation response that the sender is a FRIEND, then the Redirector 202 passes the email to the SMTP Receive Manager, at block 208, which performs its administrative function of checking the SUBSCRIBER's status and storing the email in ASM 209, which is the SUBSCRIBER'S Inbox. The user can now collect their email from the ASM Inbox (using standard Internet protocols such as POP3 or IMAP4) through the user email client 101 on their computer. Their email is 100% spam-free, since all email from senders not recognized by the system as authorized has been rejected. The SMTP Receive Manager 208 is also configured to log the information of receipt of the email from a FRIEND and send it to the ASL 203b for further analysis, as indicated at block 210.

Users send email composed on and sent from the email client 101 via standard SMTP protocols to the ISP's email server. The ISP's SMTP server is responsible for providing users with email addresses within the system, and sending users' email to the recipients' email

addresses on the Internet 103. In the SPAMKAPU invention system, an SMTP Send Manager 212 is provided to intervene in the usual send email process. The SMTP Send Manager 212 copies header information from all outgoing email and sends the data to the ASL Manager 211, then sends the email on to its intended destination. The ASL Manager 211 performs one of the key functions in the invention system. It analyzes the header data from sent email and data from other data sources 213 maintained by the ISP email server system, such as email logs and user-supplied lists. On the basis of its analysis routines (to be described in further detail below), the ASL Manager 211 checks, populates, and updates the SPDB Database and ASL List with the email addresses and other data on senders authorized to send email to the SUBSCRIBERS. The SPAMKAPU system also includes User Maintenance Modules (UMM) 214 which allows the user to interact with and upload user information to SPAMKAPU for further customization of SPAMKAPU's email operations for the user.

Referring to **FIGS. 3A and 3B**, a standard SMTP send email process (Prior Art) is shown compared to a modified send email process used in the present invention. In the standard send email process, in **FIG. 3A**, email sent from the user's email client to the ISP's email server may be pre-processed, such as checking for correct syntax, alias expansion, etc., and to identify the list of recipient email addresses (could be 1 or more). The server email manager gets each recipient email address in turn and attempts to establish a connection to the destination SMTP server and verify if the recipient email address is proper. If negotiation is unsuccessful, an error message is returned to the sending SMTP server. If negotiation is successful, the sending server sends the message body to the destination server and performs a proper "close connection" operation. In the modified send email process of the invention, in **FIG. 3B**, the email sent from the client is pre-processed, receipient(s) are indentified, and connection(s) with the destination server(s) are attempted as usual. Upon successful negotiation, the SPAMKAPU SMTP Send Manager 212 copies the successful recipient email address(es) and sends the data to the ASL Manager 211. On the assumption that the SUBSCRIBER authorizes email to be received from any person the SUBSCRIBER has sent email to, the proper email addresses of persons to whom the SUBSCRIBER has sent email are added to the ASL List of persons authorized to send email to the SUBSCRIBER. The sent email data can be used in further analyses by the ASL Manager, e.g., to upgrade a person's

authorized status from temporary to permanent if more than a threshold number of email is sent by the SUBSCRIBER to the same person.

Referring to **FIGS. 4A and 4B**, a standard SMTP receive email process (Prior Art) is shown compared to a modified receive email process used in the present invention. In the standard receive email process, in **FIG. 4A**, email is received by the SMTP server from sender sources on the Internet and the server stores the email in the user's Inbox. In the modified receive email process of the invention, in **FIG. 4B**, the received email is subjected to processing by the Redirector 202 to determine if the sender's address is that of an authorized person on the ASL List. If authorized, the SMTP server stores the email in the user's Inbox after the SMTP Receive Manager 208 captures the sender's address on the email in the address log step 210 to be sent to the ASL Manager 211. Even though the sender is already on the ASL authorized persons list, the received email data can be used in further analyses by the ASL Manager, e.g., to upgrade a persons authorized status from temporary to permanent if email from that person is received on an ongoing basis and has not been changed by the user.

In **FIG. 5**, a process flow diagram illustrates the operation of the Spam Processor 203. At block 501, a request from the calling routine, here Redirector 202, seeks validation whether a received email is from an authroized sender. The request identifies the parameters who the email is FROM and who it is sent TO. The Spam Processor 203 uses the TO address to lookup that user's ASL list 203b in the SPDB Database 203a, as indicated at block 502. The lookup procedure follows a loop 503 of reading the next ASL record on the user's ASL list, checking for a match to the email FROM address (authorized person), reading the next record if there is no match of the current record, executing the match condition by issuing a TRUE value if found, otherwise returning for the next record, as indicated at block 504. At block 505, if a TRUE VALUE is issued, then at block 505 the action is taken of setting the output value to FRIEND, ~~otherwise if no TRUE value is issued after the entire list has been processed, the action is taken of setting the output value to SPAMMER~~. At block 506, the returned value is sent as a message to the calling routine, i.e., the Redirector 202. ~~If the returned value is SPAMMER, a standard error message is included~~. As a default option, if no ASL list is found for the user, the system returns the value FRIEND, as indicated at block 507, in order to allow the email to be accepted as a temporary condition until an

or SPAMMER PK 8/24/00
as specified by the ASL table (described in further detail below) along with any appropriate error message(s) to be returned. PK 8/24/00 PK 8/24/00

ASL list can be established for that user. The request processing routine can be implemented using industry standard PERL programming syntax and incorporating a PERL interpreter to execute the processing rules.

In FIG. 6, a process flow diagram illustrates the detailed operation of the Web-Based Messenger (WBM) routine for handling email rejected by the Redirector 202 (see FIG. 2). Preferably, the WBM process is implemented via interaction with a rejected sender at a separate Web site address. In Phase 1, corresponding to step 204 in FIG. 2, the WBM process is initialized at block 601 by the ASL rule returning a value for rejecting an email as sent from a SPAMMER by the Redirector 202. At block 602, a unique ID number is ~~assigned to the email~~ CREATED in the WBM database and a given expiration date is set, e.g., 48 hours. At block 603, a return message is added along with the unique ID code to the body of the SPAMMER's email and sent back to the sender's email address in order to notify the SPAMMER to go to the WBM web page if they wish to follow through with contacting the SUBSCRIBER. The WBM then waits for the SPAMMER to go to the WBM site to complete the process, referred to as Phase 2. At block 604, the SPAMMER accesses the WBM web site and agrees to the displayed terms and conditions of usage. At block 605, the WBM process verifies that the time for response for the email corresponding to the ID number has not expired. The WBM then follows a test procedure to ensure that the responding SPAMMER is not being implemented by a mechanical program. For example, at block 606, a word stylized in non-standard font can be displayed as a graphic image, and at block 607 the SPAMMER is prompted to type the word that appears in the graphic. A mechanical program would not be able to read a graphic image of a word in unrecognizable font. At block 608, if the WBM process determines that a correct word has been typed, the SPAMMER's status is upgraded to SUSPECT on the user's ASL list. At block 609, the WBM process presents a form to enable the SUSPECT to enter a short message to be sent to the SUBSCRIBER. For example, the SUSPECT can ask the SUBSCRIBER to make sure the anti-spam control has been updated to allow email. At block 610, the email and message is sent, by routing directly to the ASM email box for the SUBSCRIBER, along with modification of the header to include a codeword or flag, e.g., adding the word "contact:" to the subject line. The codeword can be discerned in the ASM email logging step 210 in FIG. 2, in order to differentiate the redirected email from other email determined to be authorized email. At block 611, the SUBSCRIBER can now read the SUSPECT's email. If the SUBSCRIBER

RK 8/24/00

sends a reply to the email, the SUSPECT's status may be automatically upgraded to FRIEND, or the SUBSCRIBER may upgrade the status to FRIEND manually by interaction with the ISP email server through the UMM 214. At block 612, if the SUBSCRIBER determines that the email is from someone whose email should be rejected without a WBM error reply option, the SUBSCRIBER may optionally downgrade the status permanently to SPAMMER through the UMM 214.

Referring to **FIG. 7A**, a block diagram illustrates a standard SMTP send-receive email handling process (Prior Art), as compared to **FIG. 7B** which shows a modified Redirector process for handling received email. In the standard process, the Sender-SMTP 701 requests connection to the Receiver-SMTP 702, which accepts the connection if available. The Sender SMTP then performs the task in its Send Email loop of sending the recipient's email address. At block 703, the Receiver-SMTP confirms or denies whether the recipient exists or whether it has authority to process email for this user. If confirmed, the Sender-SMTP sends the message body and marks the end of the message. At block 704, the Receiver-SMTP receives the message body and sends it to the email box of the recipient (or recipients if the message is sent to more than one recipient at that SMTP server address.

In **FIG. 7B**, the Sender-SMTP 701 and Receiver-SMTP 702 perform their usual establishing of a connection and check for valid receipient e-mail address. However, in this modified process implemented in conjunction with the Spam Processor 705, the sender's FROM address is stored by the Spam Processor for later use, as indicated at block 706. At block 707, the sender's FROM address and the recipient's TO address are sent to the Spam Processor 705, by a request for validation by the Redirector as described previously. At block 708, after checking the recipient's ASL list to determine whether the sender is authorized, the Spam Processor can return a response of FRIEND or a response of SPAMMER with an accompanying error message. If the response is FRIEND, an output is sent to the Sender-SMTP confirming that the email can be received, and the email is sent to the Receiver-SMTP as usual. At block 709, the Receiver-SMTP puts the email in the recipient's email box and, if desired, can include a message noting that the sender was identified on the ASL list as a friend. If the response is SPAMMER, then an error message is returned to the Sender-SMTP that the recipient does not exist or the Recipient-SMTP is not authorized to accept the email. Optionally, the Receiver-SMTP may send the email through the

WBM process, as described previously (indicated at block 710), if the response from the Spam Processor indicates that the status of the sender is an unknown sender (as opposed to having the confirmed status of SPAMMER).

In FIG. 8, a schematic diagram illustrates the structure and operation of the ASL Manager, previously described as component 211 with respect to FIG. 2. The ASL Manager preferably is structured to have an ASL On-Demand Processor 801 and an ASL Scheduler Processor 802, both of which interact with an ASL Rules Processor 803, which also exchanges data with the Spam Processor Database (SPDB) 203a. Email addresses sent to and received from the SMTP Send Manager 212 and SMTP Receive Manager 208 are processed by the ASL On-Demand Processor 801 which executes the appropriate rules in conjunction with the ASL Rules Processor 803. Content from a variety of other sources, including compatible third party plug-ins, can also be processed to create, populate, and update the ASL Lists stored in the SPDB 203a. For example, content may be received from a "Drag and Drop Manager" for conveniently handling user address inputs while working with the email client, user address inputs from Web sites while working with an associated browser, addresses added by the user to a desktop contact manager, such as the Microsoft Outlook ™ Address Book, or other contact lists, and other address inputs generated by third party software that can operate with the user's client programs.

The ASL Scheduler Processor 802 is used to process tasks on a scheduled basis for various analysis and maintenance functions. This allows a very rich examination of the SUBSCRIBER's ASL list, mail log, and other data files, to continually refine the "authorized senders" list for accuracy and relevance. For example, the processor functions can include: an ASL Mail Log Analyzer for analyzing the ASL Mail Log database 803a of the SUBSCRIBER's received and sent emails; an Expiration Date Analyzer for setting and enforcing expiration dates for authorized senders to be re-authorized; a Low Volume Analyzer for downgrading or eliminating the authorization status of senders with whom the SUBSCRIBER communicates very infrequently; a High Volume Analyzer for upgrading or permanently marking the authorization status of senders with whom the SUBSCRIBER communicates very frequently; a Fuzzy Logic Analyzer for making qualitative decisions as to FRIEND or SPAMMER status based on a variety of factors; and other

Third Party Analyzers for analyzing data generated by third party plug-ins and programs to refine the ASL list.

The ASL Rules Processor 803 contains the rules (in an ASL Manager Rules Database) that determine how to add, update or modify the ASL Lists maintained in the SPDB Database 203a. The Rules Processor can have an architecture that readily accepts and interoperates with third party databases 803b and applications programs 803c in order to harness the collective power of developers in the network communications indsutry to continually improve and extend the SPAMKAPU system's feature set. The ultimate result of this architecture is to enable the creation of a very richly detailed ASL database which goes beyond even the total elimination of spam email into other or future needs of users for the dynamic and intelligent handling of email.

In **FIG. 9**, a detailed implementation is illustrated of examples of processing of email send/receive and user contact data into specific forms of actions taken by the ASL Manager. The basic process flow consists of: Step 901 of looping through each line of an ASL list (called a Table) comparing the FROM address captured from an incoming email for a match; Step 902 of determining whatever condition or status flag has been set for the matched entry, then executing the corresponding condition rule as maintained on the Condition Table, resulting in return of a Return Value; and Step 903, based on the Return Value, executing the corresponding action rule as maintained on the Action Table, and exiting with a Final Return Value from this action. To follow one example through this process flow, Step 901 finds a FROM match of the sender address john@home.com, Step 902 notes the expiration date condition "before 12/1/2003" and executes the "before" condition on the Condition Table to return a value of "True" if today's date isless than the indicated expiration date, and Step 903 notes that the sender status action (if condition is True) is "friend" and executes the "friend" action on the Action Table to return a Final Return Value of FRIEND (no parameters needed) as the validation response of the Spam Processor.

The specific programming syntax or execution logic of the ASL Manager rules processing may be varied in any suitable manner depending on the developer of the Spam Processor application. The following examples of some options for ASL Manager actions illustrate a wide range of approaches that may be used:

MATCHING AN EMAIL ADDRESS OR ADDRESS PATTERN:
(a) Default: exact match
(b) A specific email address: john@company.com
(c) UNIX Standard wildcard matching:
*.microsoft.com = anything from "Microsoft.com"
microsoft = anything with microsoft in it
*.mil = any email from the military
(d) Matching any known "blackhole list" by using a %BLACKHOLE% symbol.

USING A CONDITIONAL AND PARAMETERS TO EXECUTE IF THE MATCH IS TRUE

USING A SECONDARY ACTION AND PARAMETERS TO PERFORM IF THE CONDITIONAL IS TRUE.

USING THE LAST DATE THE SUBSCRIBER SENT EMAIL TO THIS ADDRESS

USING THE LAST DATE THIS ADDRESS SENT EMAIL TO THE SUBSCRIBER

USING DATE THE RECORD WAS CREATED

EXAMPLES OF CONDITIONALS THAT CAN BE USED:
(a) Expiration dates: use a given address until 2/12/2004
(b) Date ranges: use a given address from 4/1/2004 to 5/2/2004
(c) Specific recurring times: first week of every month but no other time, e.g., newsletter@magazine.com acceptable during 1st week of each month.
(d) A link to external software designed to allow for additional user-defined criteria; this allows for third party applications

EXAMPLES OF MESSAGES THAT MAY BE INVOKED BY A GIVEN SECONDARY ACTION
(a) Standard "error"
(b) Custom with variable substitution in the message body, e.g.:
%username% is substituted with the sender's email address
%subid% is the ID code of the subscriber
%date% is today's date
(c) "hello %username% you have been identified as spam, go to http://www.spamkapu.com/subscriber=%subid% and if you're really human we'll let you in.
(d) Custom text: "All email addresses from America Online are unconditionally rejected"
(e) Send a given message in the error response.
(f) Send a given message as an email.

(g) Open a file and email its contents
(h) Open a file and send its contents as an error reponse.
(i) Set the sender's status to SPAMMER or FRIEND
(j) Create a unique ID that will expire after a short time period (24-48hrs). This id can be used by the SUSPECT to access the WBM and become a CONTACT.
(k) Give SMTP default error message
(l) Link and execute external software designed to allow for additional user-defined actions; this allows for third party applications.

In summary, the present invention provides a spam email rejection method which analyzes the sender address of incoming email and determines whether it is to be rejected or accepted depending upon managed lists of authorized senders. This is a significant departure from existing anti-spam processing systems which accept all email and attempts to filter out only those that have sender addresses recognized as those of known spammers. The invention method does not filter out unauthorized email, rather it rejects all email unless authorized. The ASL Manager in the system captures and analyzes sender and recipient usage patterns for outgoing and incoming email in order to refine the "authorized senders" lists. The analysis of this data provides a rich foundation for rules-based decisions as to which sender addresses are considered SPAMMER and which are not. This data creates an "authorized sender" list of FRIENDS, as opposed to a list of known SPAMMERS, thereby ensuring that no unsolicited or uninvited email will ever pass through to the SUBSCRIBER's eamil box.

It is understood that many other modifications and variations may be devised given the above description of the guiding principles of the invention. It is intended that all such modifications and variations be considered as within the spirit and scope of this invention, as defined in the following claims.

I CLAIM:

1. A system for eliminating unauthorized email sent to a user on a network comprising:

(a) an email client for allowing the user to receive email sent on the network addressed to a unique email address of the user,

(b) an email-receiving server connected between the network and the email client for receiving email addressed to the unique email address of the user, said email-receiving server having an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, and

(c) an email rejection module operable with the ASL module for rejecting the receipt of email addressed to the email address of the user if the email address of the sender is not one that is maintained on the ASL list for the user.

2. A system according to Claim 1, wherein the ASL module includes an ASL database for storing ASL lists of authorized sender addresses for respective subscribers of the system, a spam processor module for checking the ASL lists for matches, and an ASL manager for creating, maintaining, and updating the ASL lists.

3. A system according to Claim 2, further comprising a redirector module operable with the ASL module for receiving an email-sending message designating the sender's FROM address and intended recipient's TO address, for sending a request for validation to the spam processor module to determine whether the sender's FROM address matches any authorized sender address maintained on the ASL list corresponding to the TO address of the intended recipient, for accepting the email if a match to an authorized sender address is found, and for rejecting the email if no match to an authorized sender address is found on the ASL list.

4. A system according to Claim 3, further comprising a web-based messaging (WBM) module to which email rejected by the redirector module is redirected and which sends a message to the address of the sender of the rejected email notifying the sender to confirm that the

sender is a legitimate sender of email to the intended recipient.

5. A system according to Claim 4, wherein the WBM module includes a separate web site to which the notified sender can log on and confirm that the sender is a legitimate sender of email through an interaction procedure which can only be performed by a human.

6. A system according to Claim 5, wherein the interaction procedure includes a display of a graphic image of a word in a non-standard font, and an input for the sender to enter in a word corresponding to the graphic image of the word, whereby the system can confirm that the interaction procedure is not performed by a mechanical program.

7. A system according to Claim 4, wherein once the sender is confirmed as a legitimate sender of email to the intended recipient user, the WBM module sends the email to the user's email box with a code that indicates that the email was rejected by the redirector module but confirmed as legitimate by the WBM module.

8. A system according to Claim 3, further comprising an email-receiving manager for capturing FROM and TO addresses of email accepted by the redirector module and sending the data to the ASL manager for later analysis.

9. A system according to Claim 2, further comprising an email-sending manager for capturing FROM and TO addresses of email sent from the email client and sending the data to the ASL manager for later analysis.

10. A system according to Claim 2, wherein the ASL manager further includes a rules processor for processing predefined address capture rules for updating the ASL lists using data from an email address source selected from the group of email address sources consisting of: received email; sent email; user inputs to email service functions on the email client; inputs from user browsing of web sites; user desktop organizer and other contact lists; and third party address program inputs.

11. A system according to Claim 2, wherein the ASL manager further comprises a rules processor for processing predefined analysis rules for updating the ASL lists using data from an analysis source selected from the group of analysis sources consisting of: user email log analysis; expiration date analysis; low/high email volume analysis; fuzzy logic analysis; and third party data analysis.

12. A system according to Claim 2, wherein the ASL manager maintains the ASL lists designating a sender-address status selected from the group of sender-address statuses consisting of: always authorized as a friend; authorized as a friend over a date range; authorized as a friend before an expiration date; always rejected as a spammer; rejected as a spammer matching a black list; and rejected as a spammer sent with an error message.

13. A method for eliminating unauthorized email sent to a user on a network comprising the steps of:

(a) receiving email addressed to the unique email address of the user,

(b) maintaining an authorized senders list (ASL list) of email addresses of external users authorized to send email to the user, and

(c) rejecting the receipt of email sent to the email address of the user if the email address of the sender is not one maintained on the ASL list for the user.

14. A method according to Claim 13, further comprising the step of redirecting the rejected email to a web site for sending a message to the sender of the rejected email notifying the sender to confirm that the sender is a legitimate sender of email to the intended recipient.

15. A method according to Claim 14, further comprising the step of performing an interaction procedure at the web site with the notified sender which can only be performed by a human.

16. A method according to Claim 13, wherein said ASL list maintaining step includes updating the ASL lists using data captured from any of the following sources: received email; sent email; user inputs to email service functions; inputs from user browsing of web sites;

user desktop organizer and other contact lists; and third party address program inputs.

17. A method according to Claim 13, wherein said ASL list maintaining step includes updating the ASL lists using data obtained from analysis of any of the following factors: user email log analysis; expiration date analysis; low/high email volume analysis; fuzzy logic analysis; and third party data analysis.

18. An email server system for eliminating unauthorized email sent via a network to the server addressed to a unique email address for a user of the system comprising:

(a) an authorized senders list (ASL) module which maintains an ASL list of email addresses of senders authorized to send email to the user, and

(b) an email rejection module operable with the ASL module for rejecting the receipt of email addressed to the email address of the user if the email address of the sender is not one that is maintained on the ASL list for the user.

19. An email server system according to Claim 19, wherein the ASL module includes an ASL database for storing ASL lists of authorized sender addresses for respective subscribers of the system, a spam processor module for checking the ASL lists for matches, and an ASL manager for creating, maintaining, and updating the ASL lists.

20. An email server system according to Claim 19, further comprising a redirector module operable with the ASL module for receiving an email-sending message designating the sender's FROM address and intended recipient's TO address, for sending a request for validation to the spam processor module to determine whether the sender's FROM address matches any authorized sender address maintained on the ASL list corresponding to the TO address of the intended recipient, for accepting the email if a match to an authorized sender address is found, and for rejecting the email if no match to an authorized sender address is found on the ASL list.

ABSTRACT OF THE DISCLOSURE

A system for eliminating unauthorized email sent to a user on a network employs an email-receiving server connected between the network and the user's email client for receiving email addressed to the user and rejecting those in which the sender address does not match any of sender addresses maintained on an "authorized senders" list (ASL list). The ASL lists are maintained by an ASL manager in an ASL database operable with a spam processor module. A redirector module rejects the email if, upon sending a request for validation to the spam processor module, the sender's address does not match any authorized sender address on the ASL list. Email rejected by the redirector module is redirected to a web-based messaging (WBM) module which sends a message to the sender to confirm that the sender is a legitimate sender of email to the intended recipient. If the sender logs on to confirm their status, the WBM module executes an interaction procedure which can only be performed by a human, in order to ensure that the confirmation procedure is not performed by a mechanical program. The ASL manager maintains the ASL lists based upon sender address data collected from various sources and analyses of various email usage factors, including sent email, received email, contact lists maintained by the user, user preference inputs, third party programs, etc.

FIG. 1A (PRIOR ART)

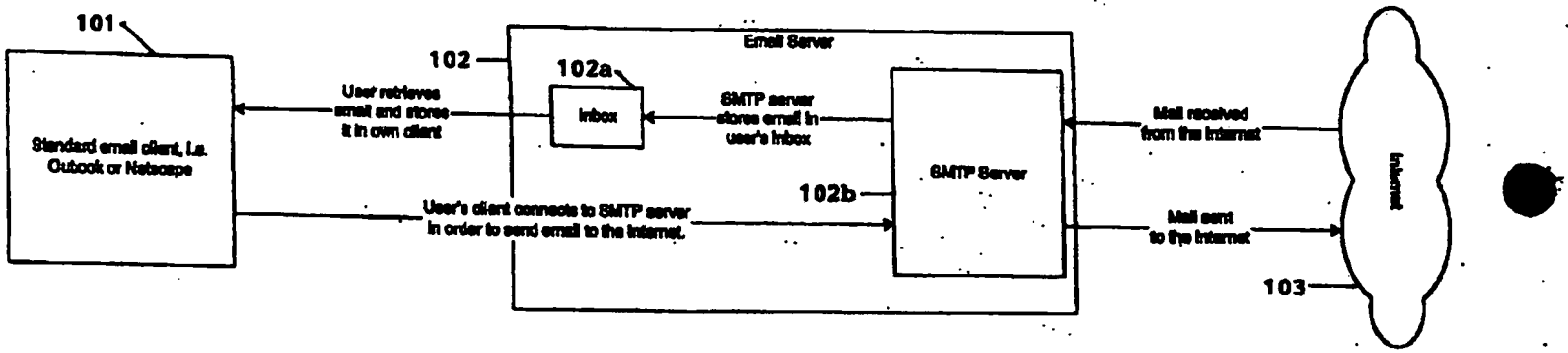


FIG. 1B

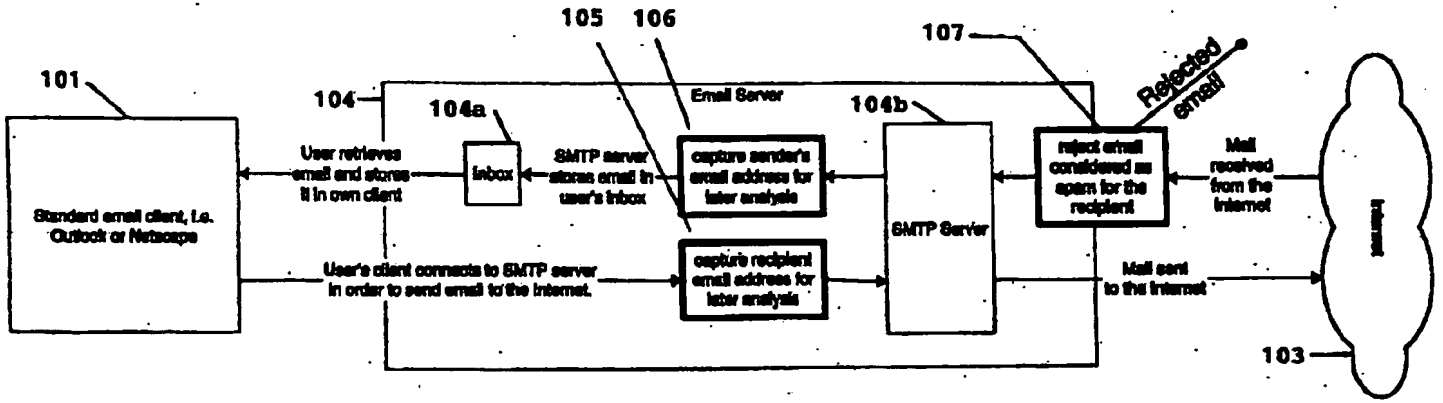


FIG. 2

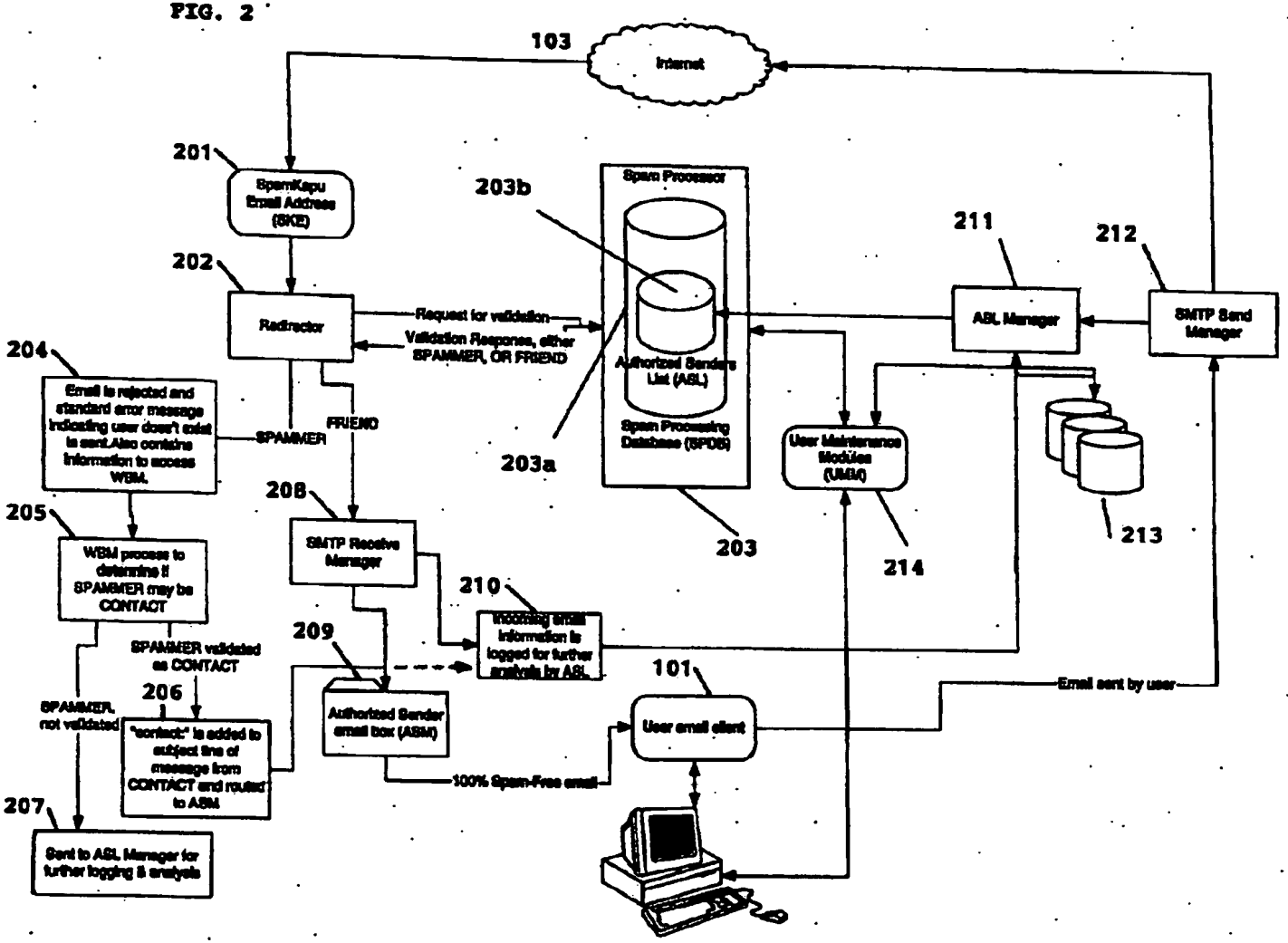

103
Internet
201
SpamKapu Email Address (SK2)
202
Redirector
Request for validation
Validation Response, either SPAMMER, OR FRIEND
204
Email is rejected and standard error message indicating user does't exist is sent.Also contains information to access WBM.
SPAMMER
FRIEND
205
WBM process to determine if SPAMMER may be CONTACT
SPAMMER validated as CONTACT
206
SPAMMER, not validated
"contact:" is added to subject line of message from CONTACT and routed to ASM
207
Sent to ASL Manager for further logging & analysis
208
SMTP Receive Manager
209
Authorized Sender email box (ASM)
210
Incoming email information is logged for further analysis by ASL
203b
Spam Processor
Authorized Senders List (ASL)
Spam Processing Database (SPDB)
203a
203
211
ASL Manager
212
SMTP Send Manager
User Maintenance Modules (UMM)
214
213
101
User email client
Email sent by user
100% Spam-Free email

FIG. 3A (PRIOR ART)

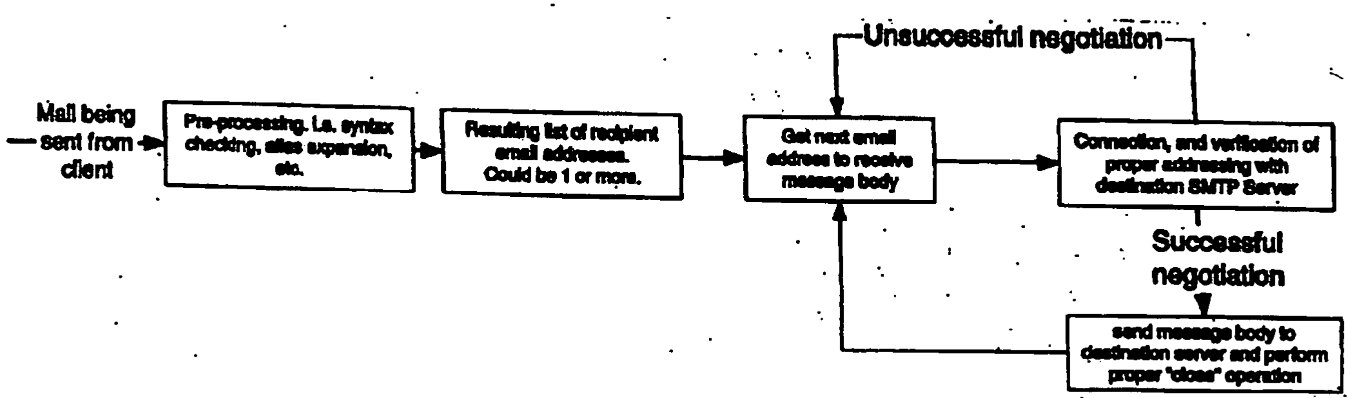


FIG. 3B

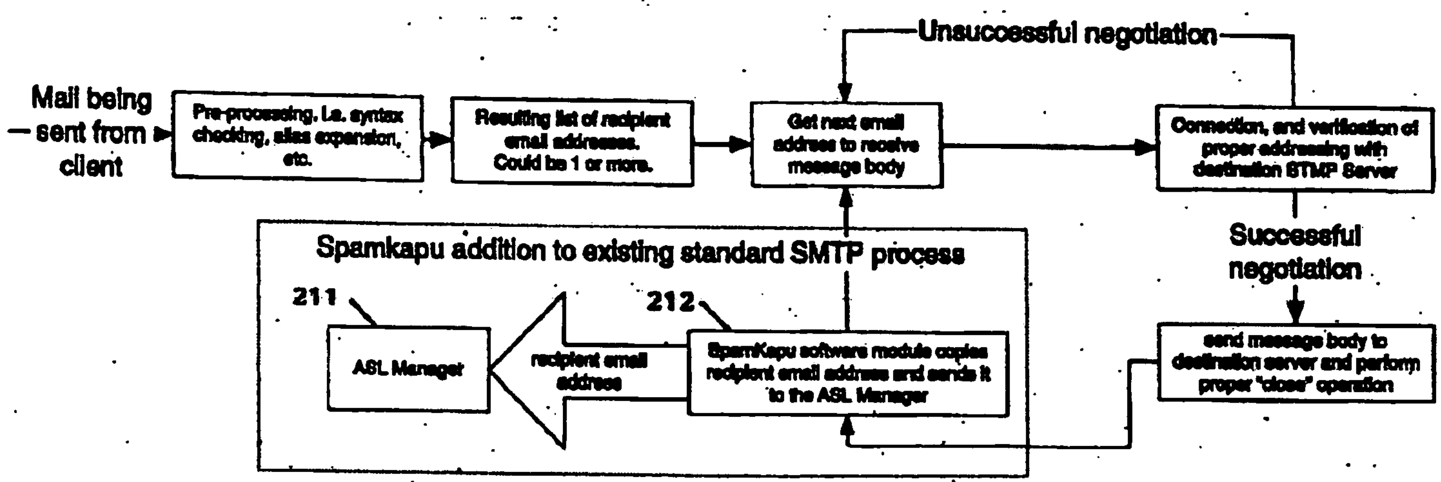


FIG. 4A (PRIOR ART)

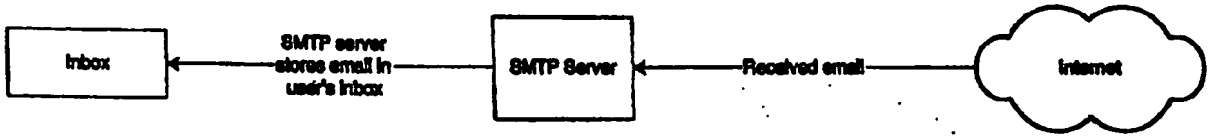


FIG. 4B

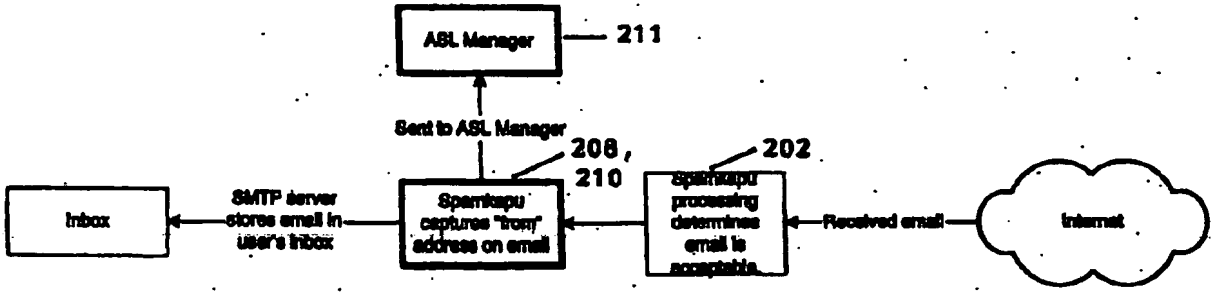


FIG. 5

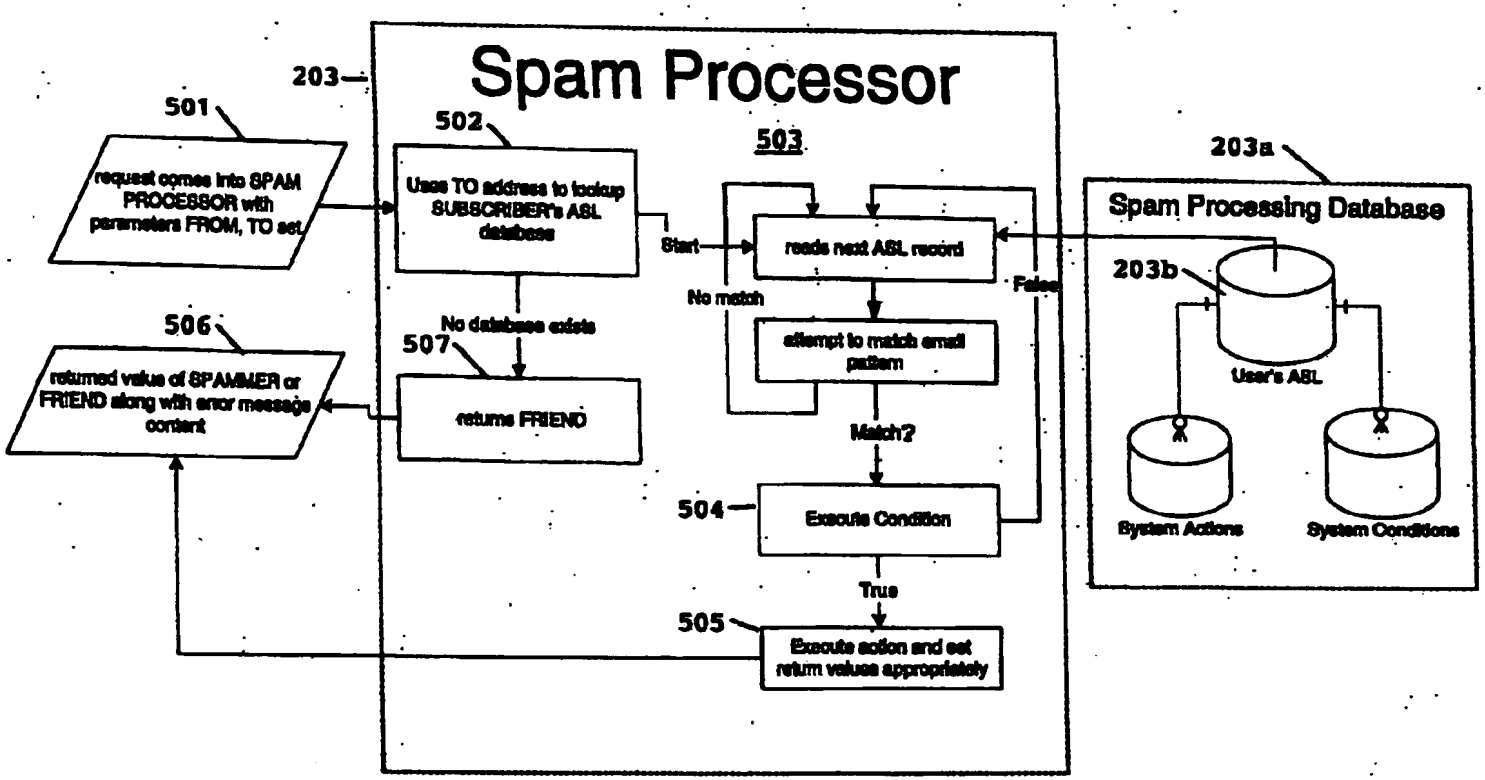
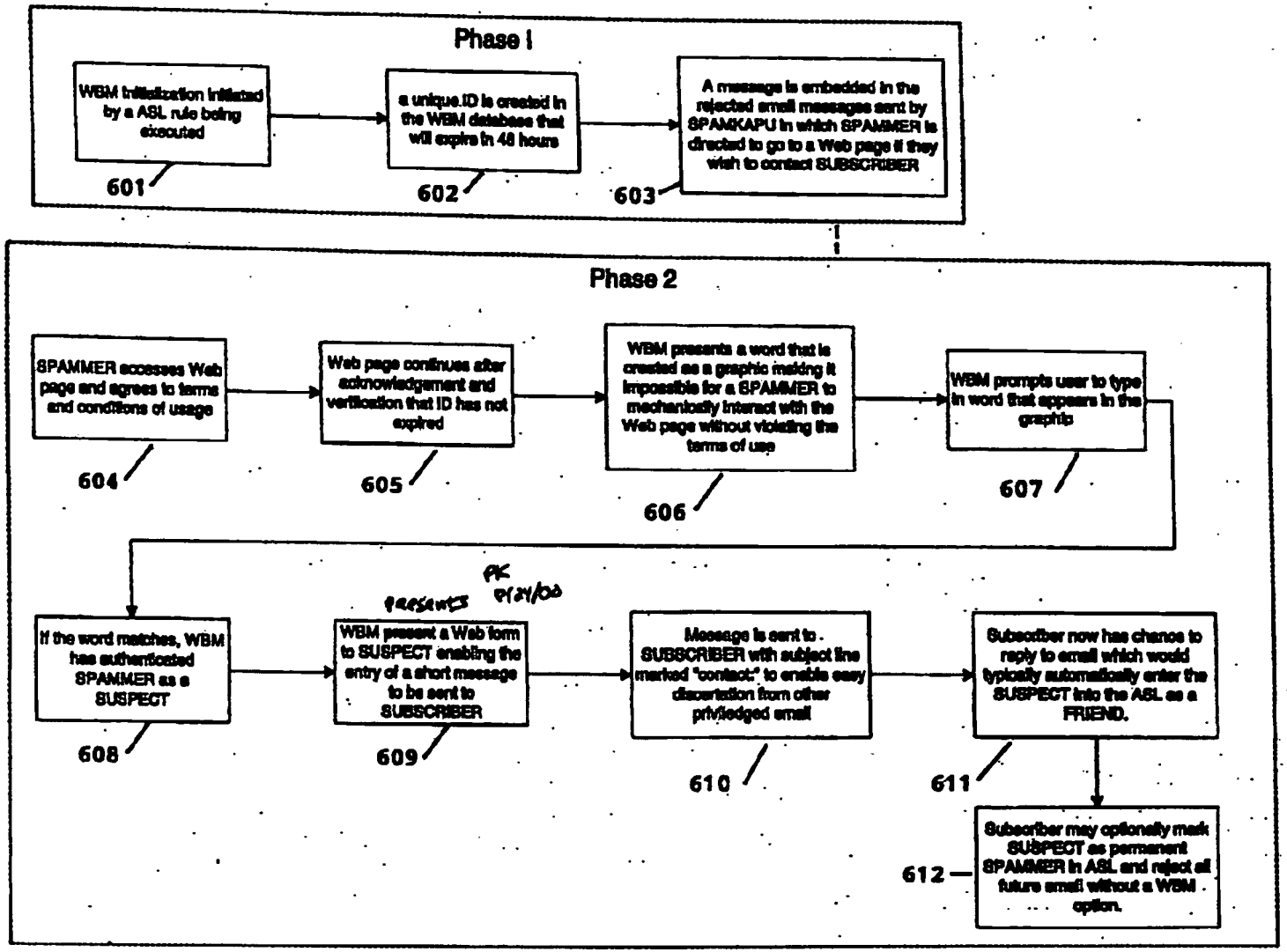

Spam Processor
203
501
request comes into SPAM PROCESSOR with parameters FROM, TO set
502
Uses TO address to lookup SUBSCRIBER's ASL database
503
Start
reads next ASL record
No match
attempt to match email pattern
Match?
False
504
Execute Condition
True
505
Execute action and set return values appropriately
No database exists
507
returns FRIEND
506
returned value of SPAMMER or FRIEND along with error message content
203a
Spam Processing Database
203b
User's ASL
System Actions
System Conditions

FIG. 6


Phase I
WBM initialization initiated by a ASL rule being executed
601
a unique ID is created in the WBM database that will expire in 48 hours
602
A message is embedded in the rejected email messages sent by SPAMKAPU in which SPAMMER is directed to go to a Web page if they wish to contact SUBSCRIBER
603
Phase 2
SPAMMER accesses Web page and agrees to terms and conditions of usage
604
Web page continues after acknowledgement and verification that ID has not expired
605
WBM presents a word that is created as a graphic making it impossible for a SPAMMER to mechanically interact with the Web page without violating the terms of use
606
WBM prompts user to type in word that appears in the graphic
607
If the word matches, WBM has authenticated SPAMMER as a SUSPECT
608
WBM present a Web form to SUSPECT enabling the entry of a short message to be sent to SUBSCRIBER
609
Message is sent to SUBSCRIBER with subject line marked "contact:" to enable easy discertation from other privledged email
610
Subscriber now has chance to reply to email which would typically automatically enter the SUSPECT into the ASL as a FRIEND.
611
Subscriber may optionally mark SUSPECT as permanent SPAMMER in ASL and reject all future email without a WBM option.
612

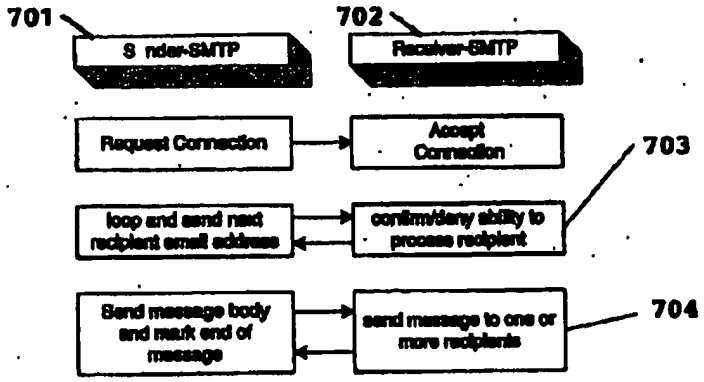

701
S nder-SMTP
702
Receiver-SMTP
Request Connection
Accept Connection
703
loop and send next recipient email address
confirm/deny ability to process recipient
Send message body and mark end of message
send message to one or more recipients
704

FIG. 7A

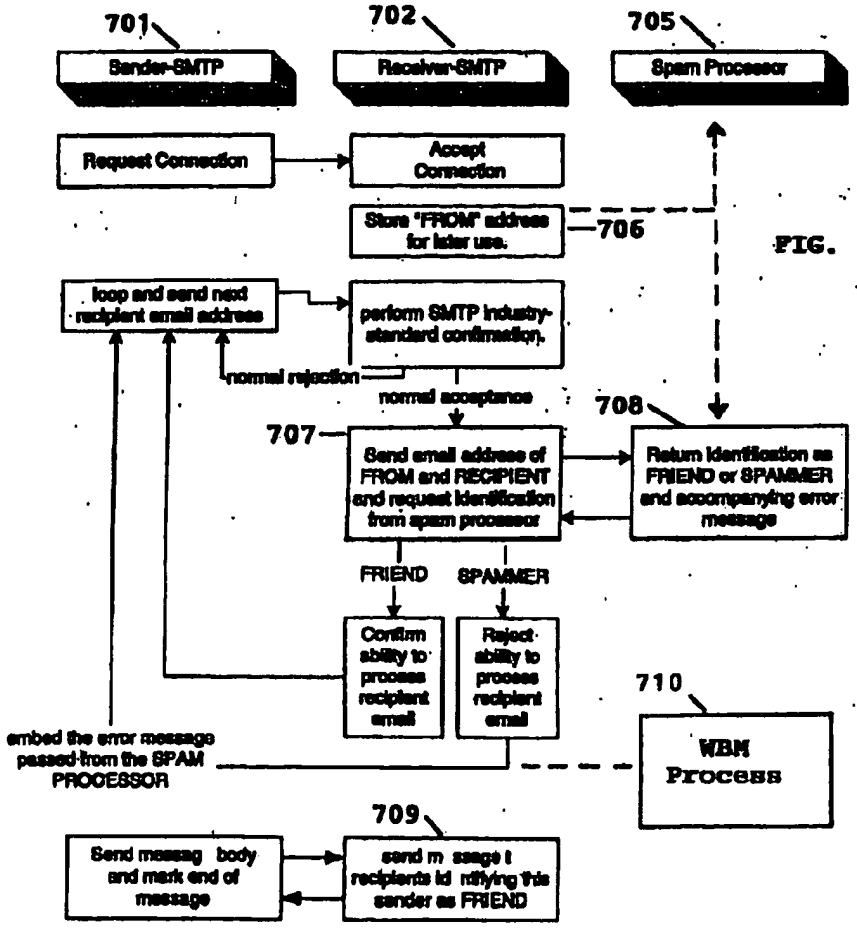

701
Sender-SMTP
702
Receiver-SMTP
705
Spam Processor
Request Connection
Accept Connection
Store "FROM" address for later use.
706
loop and send next recipient email address
perform SMTP industry-standard confirmation.
normal rejection
normal acceptance
708
707
Send email address of FROM and RECIPIENT and request identification from spam processor
Return identification as FRIEND or SPAMMER and accompanying error message
FRIEND
SPAMMER
Confirm ability to process recipient email
Reject ability to process recipient email
710
embed the error message passed from the SPAM PROCESSOR
WBM Process
709
Send messag body and mark end of message
send m ssage t recipients id ntifying this sender as FRIEND

FIG. 7B

FIG. 8

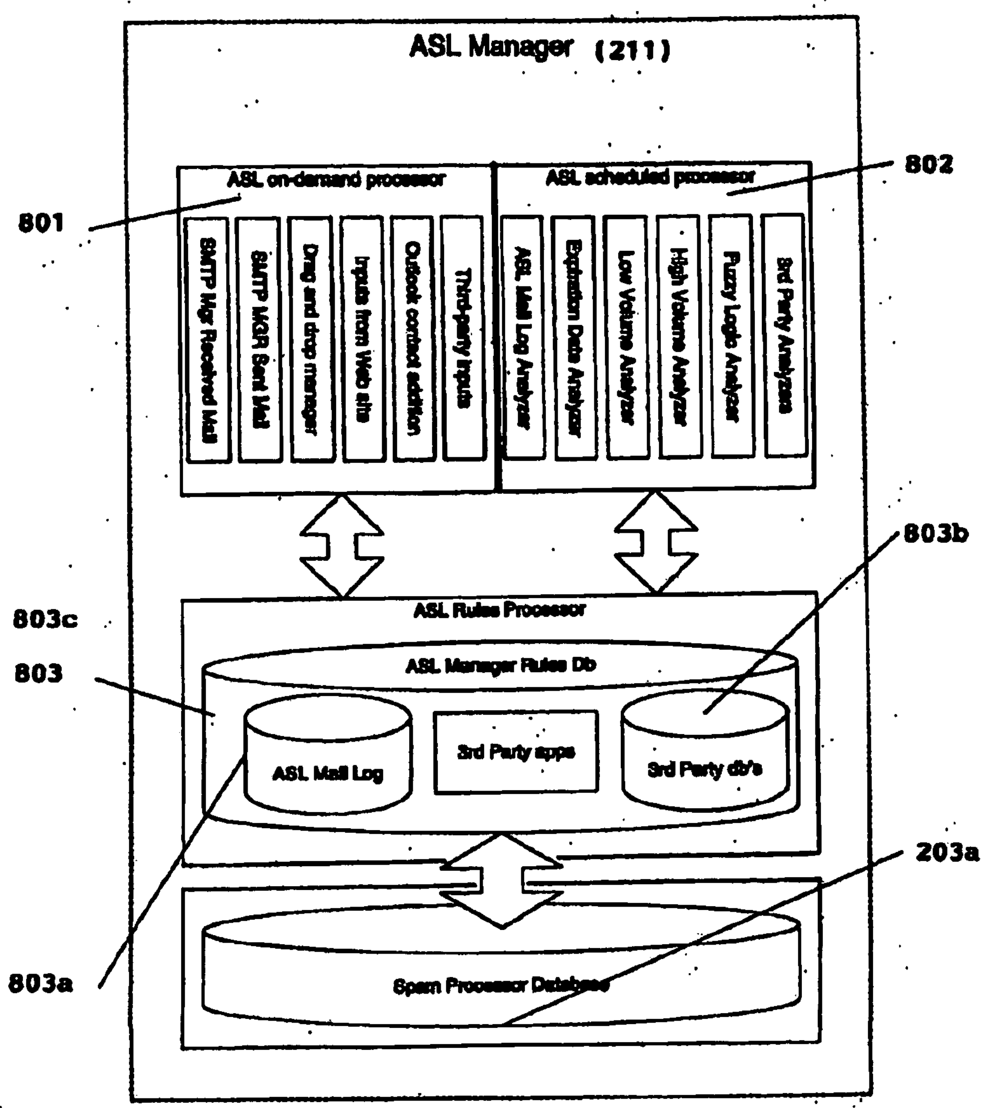
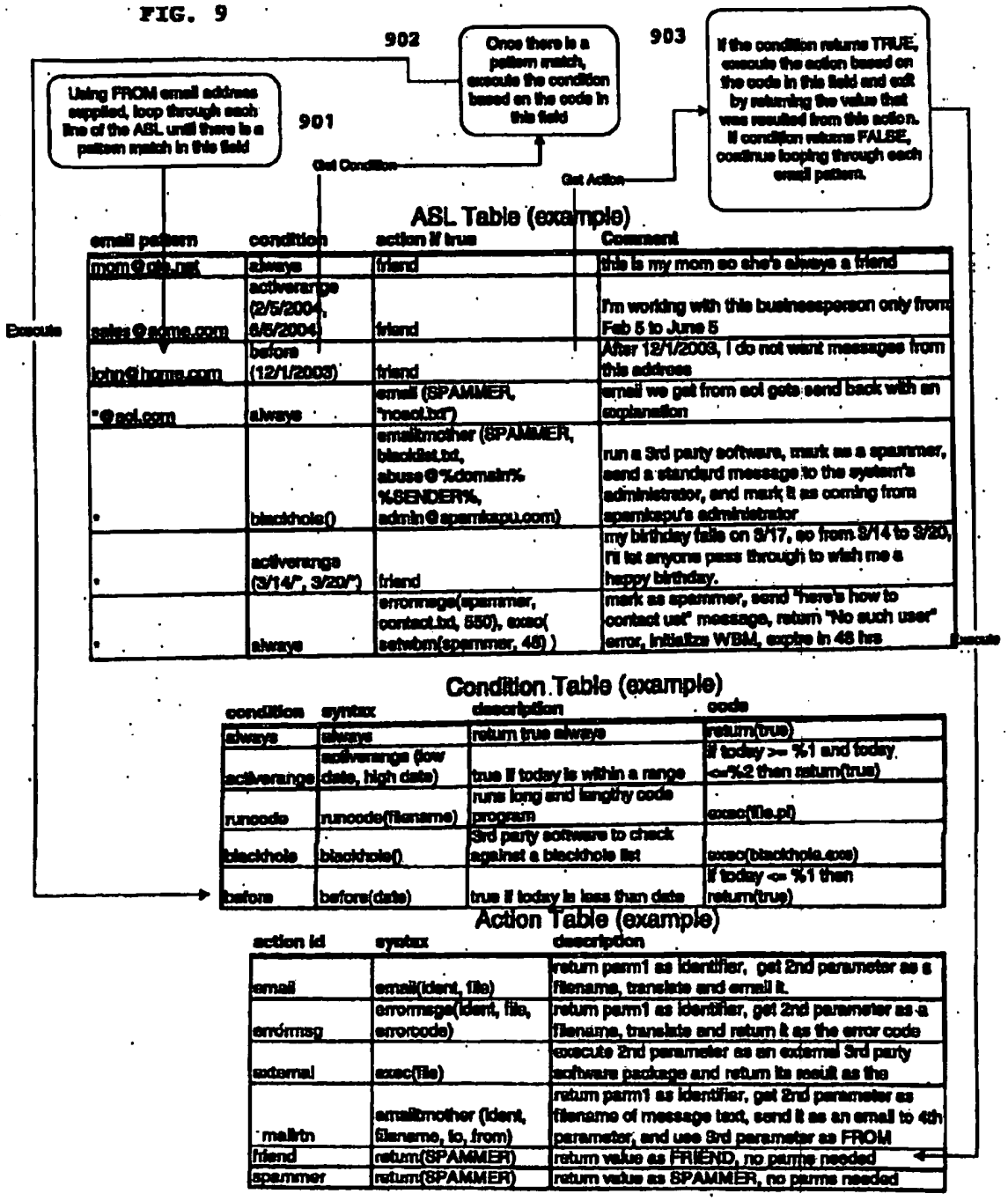

ASL Manager (211)
ASL on-demand processor
SMTP Mgr Received Mail
SMTP MGR Sent Mail
Drag and drop manager
Inputs from Web site
Outlook contact addition
Third-party inputs
ASL scheduled processor
ASL Mail Log Analyzer
Expiration Date Analyzer
Low Volume Analyzer
High Volume Analyzer
Fuzzy Logic Analyzer
3rd Party Analyzers
801
802
ASL Rules Processor
ASL Manager Rules Db
ASL Mail Log
3rd Party apps
3rd Party db's
803c
803
803b
803a
203a
Spam Processor Database

FIG. 9


902
Once there is a pattern match, execute the condition based on the code in this field
903
If the condition returns TRUE, execute the action based on the code in this field and exit by returning the value that was resulted from this action. If condition returns FALSE, continue looping through each email pattern.
Using FROM email address supplied, loop through each line of the ASL until there is a pattern match in this field
901
Get Condition
Get Action
Execute
ASL Table (example)
email pattern
condition
action if true
Comment
mom@aol.net
always
friend
this is my mom so she's always a friend
sales@some.com
activerange (2/5/2004, 6/5/2004)
friend
I'm working with this businessperson only from Feb 5 to June 5
john@home.com
before (12/1/2003)
friend
After 12/1/2003, I do not want messages from this address
*@aol.com
always
email (SPAMMER, "noaol.txt")
email we get from aol gets send back with an explanation
*
blackhole()
emailmother (SPAMMER, blacklist.txt, abuse@%domain% %SENDER%, admin@spamkapu.com)
run a 3rd party software, mark as a spammer, send a standard message to the system's administrator, and mark it as coming from spamkapu's administrator
*
activerange (3/14/*, 3/20/*)
friend
my birthday falls on 3/17, so from 3/14 to 3/20, I'll let anyone pass through to wish me a happy birthday.
*
always
errormsg(spammer, contact.txt, 550), exec(setvbm(spammer, 48))
mark as spammer, send "here's how to contact us!" message, return "No such user" error, initialize WBM, expire in 48 hrs
Execute
Condition Table (example)
condition
syntax
description
code
always
always
return true always
return(true)
activerange
activerange (low date, high date)
true if today is within a range
if today >= %1 and today <=%2 then return(true)
runcode
runcode(filename)
runs long and lengthy code program
exec(file.pl)
blackhole
blackhole()
3rd party software to check against a blackhole list
exec(blackhole.exe)
before
before(date)
true if today is less than date
if today <= %1 then return(true)
Action Table (example)
action id
syntax
description
email
email(ident, file)
return parm1 as identifier, get 2nd parameter as a filename, translate and email it.
errormsg
errormsg(ident, file, errorcode)
return parm1 as identifier, get 2nd parameter as a filename, translate and return it as the error code
external
exec(file)
execute 2nd parameter as an external 3rd party software package and return its result as the
mailrtn
emailmother (ident, filename, to, from)
return parm1 as identifier, get 2nd parameter as filename of message text, send it as an email to 4th parameter, and use 3rd parameter as FROM
friend
return(FRIEND)
return value as FRIEND, no parms needed
spammer
return(SPAMMER)
return value as SPAMMER, no parms needed